HTMOT : Hierarchical Topic Modelling Over Time

Judicael Poumay
ULiege/HEC Liege
Rue Louvrex 14, 4000 Liege, Belgium
judicael.poumay@uliege.be

Ashwin Ittoo
ULiege/HEC Liege
Rue louvrex 14, 4000 Liege, Belgium
ashwin.ittoo@uliege.be

Abstract

Topic models provide an efficient way of extracting insights from text and supporting decision-making. Recently, novel methods have been proposed to model topic hierarchy or temporality. Modeling temporality provides more precise topics by separating topics that are characterized by similar words but located over distinct time periods. Conversely, modeling hierarchy provides a more detailed view of the content of a corpus by providing topics and sub-topics. However, no models have been proposed to incorporate both hierarchy and temporality which could be beneficial for applications such as environment scanning. Therefore, we propose a novel method to perform Hierarchical Topic Modelling Over Time (HTMOT). We evaluate the performance of our approach on a corpus of news articles using the Word Intrusion task. Results demonstrate that our model produces topics that elegantly combine a hierarchical structure and a temporal aspect. Furthermore, our proposed Gibbs sampling implementation shows competitive performance compared to previous state-of-the-art methods.

1 Introduction

In the field of natural language processing (NLP), numerous methods for extracting topics from a corpus have been proposed over the years (Alghamdi and Alfalqi, 2015; Barde and Bainwad, 2017). While the seminal Latent Dirichlet Allocation (LDA) algorithm (Blei et al., 2003) paved the way for topic modeling, it lacks the ability to capture hierarchical or temporal information.

In the past, hierarchical topic models have been proposed (Paisley et al., 2015; Blei et al., 2004) that enable the extraction of topics and sub-topics organized in a tree-like structure. These models dynamically determine the appropriate number of topics and sub-topics during training and have been found to be useful in ontology learning (Zhu et al., 2017) and research idea recommendation (Wang et al., 2019).

In parallel, temporal topic models have been developed (Wang and McCallum, 2006; Nallapati et al., 2007; Song et al., 2008; Blei and Lafferty, 2006) that allow for the extraction of topics that describe events or trends occurring in a corpus. These models have been applied to tasks such as tracking trends in scientific articles (Hong et al., 2011) and events in social media (Zhou and Chen, 2013).

Combining hierarchical and temporal information in models can capture broad and detailed aspects of a corpus, benefiting applications like environment scanning (El Akrouchi et al., 2021). Hierarchical modeling yields detailed topics and subtopics for a comprehensive thematic understanding, while temporal modeling provides precise descriptions of events. This integration produces nuanced models for informed decision-making and deeper insights.

However, integrating temporal and hierarchical information in topic models remains a challenge (Nallapati et al., 2007; Song et al., 2008; Blei and Lafferty, 2006; Wang and McCallum, 2006). Many temporal models have their own structures to represent time, such as time trees or time slices, which complicates the integration with a hierarchical structure (Nallapati et al., 2007; Song et al., 2008; Blei and Lafferty, 2006). The only temporal model that does not require its own structure is ToT (Wang and McCallum, 2006), but combining time and hierarchy is still difficult due to the beta distribution used to model time lacking a known conjugate prior, making it incompatible with stochastic variational inference (SVI) used by previous hierarchical models (Wang and McCallum, 2006).

Our proposed method, Hierarchical Topic Modelling Over Time (HTMOT), jointly models topic hierarchy and temporality to leverage the strengths

Proceedings of Recent Advances in Natural Language Processing, pages 854–863
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_092

of both dimensions and to overcome the challenges associated with integrating them.

As a secondary contribution, we propose a novel implementation of Gibbs sampling based on a tree-based data structure called the *Infinite Dirichlet Tree*. This implementation is comparable to SVI in terms of speed. Our work provides a promising avenue for addressing the need for topic models that can incorporate both hierarchical and temporal information. (Wang and McCallum, 2006)

We performed our experiments using a corpus of 62k news articles and evaluated our method using the Word Intrusion task (Chang et al., 2009).

2 Related Work

We now describe previous topic modelling methods most closely related to ours. For more comprehensive reviews see Alghamdi and Alfalqi (2015) and Barde and Bainwad (2017).

2.1 Topic Modelling

The seminal LDA (Blei et al., 2003) algorithm remains the most popular topic model. It is the basis of most subsequent models. At the core of LDA is a Bayesian generative model based on Dirichlet distributions. These are used to model the document-topic and the topic-word distributions. They are learnt and optimized via an inference procedure, which enables topics to be extracted. The main weakness of LDA is that it requires the user to specify a predefined number of topics to be extracted. However, such information is usually not known in advance. Consequently, LDA requires a long model validation step to determine the number of topics.

The subsequent HDP (Teh et al., 2006) model uses Dirichlet processes (DPs) to determine the number of topics during training. Using DPs allows us to have an indefinite number of topics contrary to Dirichlet distributions. Otherwise, HDP operates similarly to LDA.

2.2 Hierarchical Topic Modelling

Methods such as LDA and HDP are only capable of extracting a flat topic structure. Hence, new methods have been developed to model topic hierarchies. By extracting topics and sub-topics, we end up with more detailed information about a corpus.

The state-of-the-art for hierarchical topic modelling is nHDP (Paisley et al., 2015). It models topic hierarchy by defining a potentially infinite tree where each node corresponds to a topic. At each branch of the tree, we exactly have the HDP model. The difference is that, when a word is assigned to a topic during training, there is a chance to go deeper in the tree based on a Bernoulli distribution. If we do go deeper, we repeat the HDP algorithm with a sub-corpus made up of the documents and tokens assigned to the selected topic.

Other topic models have been proposed to model hierarchy. hPAM (Mimno et al., 2007) proposes a directed acyclic graph structure instead of a tree to model topic hierarchy. Thus, high-level topics can share low-level topics. While this provides more precise relationships between topics, it is harder to display and navigate. LSHTM (Pujara and Skomoroch, 2012) recursively applies LDA to the sub-corpus defined by the topics of the previous LDA application. Hence, each new application of LDA provides a new depth to the topic tree. However, it requires a pre-defined set of parameters to define the shape of the final topic tree. Finally, the nCRP (Blei et al., 2004) is the predecessor of nHDP and works similarly. Nevertheless, it does not model the document-topic distribution as in nHDP. Consequently, the extracted documents do not have their own topic tree. Hence, nHDP is more powerful than LSHTM and nCRP (Pujara and Skomoroch, 2012; Blei et al., 2004) while keeping a strict tree structure contrary to hPAM (Mimno et al., 2007).

2.3 Temporal Topic Modelling

Previous works also investigated the temporality of topics. Providing information about when a topic occurred and/or how it evolved. Understanding the temporality of topics is important, especially for environment scanning where events and changes in the environment are important signals.

The ToT (Wang and McCallum, 2006) model is a modified version of LDA which incorporates temporality. Each document/word is associated with a timestamp which are used to fit a beta distribution for each topic. This beta distribution is optimized jointly as the topics are being discovered. The results show topics that are either better localized in time (events with specific dates) or with a clear evolution through time (growth/decline).

Other topic models have been proposed to model temporality. MTT (Nallapati et al., 2007) creates a tree for each topic which provides the ability to understand topics at various time scales. Specifically, deeper nodes correspond to a smaller timescale.

DTM (Blei and Lafferty, 2006) slices the corpus by periods. The first slice is processed similarly to LDA and the following slices are processed using the previous one as prior. Finally, the Dynamic Correlated Topic Model (DCTM) (Song et al., 2008) also slices the corpus in periods. However, it uses Gaussian processes and Singular Value Decomposition (SVD) instead of LDA-based techniques. The advantage of ToT is that it is non-Markovian and it models time as a continuum. Hence, ToT is the only model which does not require its own structure to model time such as slices or a binary tree. This is important if we are already building a structure for the topic hierarchy.

2.4 Topic Models Evaluation

Previous studies have used various methods to evaluate topic models, such as perplexity and coherence. However, these methods have been repeatedly shown to be uncorrelated with human judgement (Chang et al., 2009; Hoyle et al., 2021; Doogan and Buntine, 2021; Bhatia et al., 2017).

Consequently, the Word Intrusion task was proposed as an evaluation method that involves inserting an intruder word into a topic's top word list and asking annotators to identify it (Chang et al., 2009). The intruder word is selected at random from a pool of words with a low probability in the current topic but a high probability in another topic to avoid rare words. The idea is that, in good topics, it should be easy for annotators to identify the intruder word. The final score is the average classification accuracy made by humans. In (Lau et al., 2014), this task was automated with performance similar to human annotators.

3 HTMOT : Hierarchical Topic Modelling Over Time

We now describe our method for HTMOT. We begin by presenting a new type of data structure at the core of HTMOT (section 3.1). Next, we describe how temporality was incorporated into the hierarchy (section 3.2). Then, we detail our novel implementation of Gibbs sampling (section 3.3). Finally, we denote important differences between HTMOT and its predecessor (section 3.4).

3.1 Counting Words Using Infinite Dirichlet Trees

Infinite Dirichlet Trees (IDTs) are efficient tree-based data structures we developed. The name refers to the potentially infinite number of topics provided by the Dirichlet Processes, which define how they grow. The role of these trees is to model the topics, their hierarchical dependency, and temporality. Hence, these trees are optimized during the training process to serve as the final output of HTMOT.

Each node of an IDT is identified by a finite path in the tree as a sequence of node ids, starting from the root. For example, the node "root.A.B" corresponds to a sub-topic of the topic "Root.A". The nodes record word assignments (see figure 1) and the timestamps of those words (associated with the source document). Thus, each node represents a topic and defines a *topic-word* and a *topic-time distribution*.

The trees also model the hierarchical distribution of topics. Words are assigned to a final topic and to all ancestors of that topic. Hence, there are two types of word assignments : "through" and "final", respectively for the ancestor topics and final topic. This creates a hierarchical dependency between the nodes and thus a *hierarchical distribution*.

We use multiple IDTs, one for the corpus and one for each document. All words in the corpus are assigned to nodes of the corpus tree. Similarly, each document has an associated document tree recording each word of that document. Hence, combining all document trees together would yield the corpus tree. For both the corpus and document trees, each node (topic) will be assigned a different number of words. Thus, nodes differ in size which creates a distribution. Hence, the corpus tree defines a *corpus-topic distribution* and each document tree defines a *document-topic distribution*.

From the foregoing discussion, we can see that the assignment of words to the different trees defines the *topic-word, topic-time, document-topic, corpus-topic and topic-hierarchy distributions.* Hence, by simply moving words around in those trees, we can optimize all these distributions jointly. Once optimized, the trees can be used directly as output to view topics, their hierarchy and temporality for the corpus and each document.

3.2 Modelling Temporality

Temporality is modeled by associating topics with a beta distribution as in ToT (Wang and McCallum, 2006). This allows us to extract topics that describe specific events in time. Mathematically, we separate topics that are lexically similar but located at

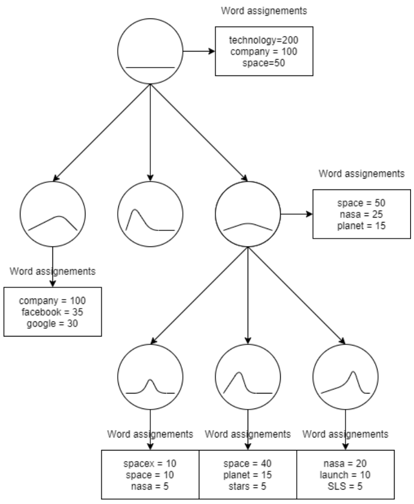

Figure 1: Example of an IDT with word assignments and time distribution (inside nodes).

Algorithm 1 Traditional Gibbs sampling

```
procedure ClassicGibbs(corpus)
    for N iterations do
        for each document in corpus do
            for each word in document do
                Sample word-topic
                Sample topic-word
                Sample document-topic
                Estimate time-topic
                Sample corpus-topic
                Sample hierarchy-topic
            end for
        end for
    end for
    Return solution
end procedure
```

different periods in time. However, applying temporality to high-level topics would split them into various periods. Each of these splits would have similar sub-topics, which would lead to an unnecessary multiplication of topics. Hence, contrary to ToT, we do not apply temporality to all topics but only deep ones. For our experiments, we choose depths of 3 or more. This allows us to extract precise topics about specific events in time at deeper levels while keeping the high-level topics intact.

The parameters of the beta distribution ρ_i^1 and ρ_i^2 are computed for a topic i based on the current timestamps assignments (associated with each word assignment). We used the method of the moment to estimate these parameters :

$$\rho_i^1 = \overline{t_i} * (\frac{\overline{t_i} * (1 - \overline{t_i})}{\sigma_{t_i}} - 1) \quad (1)$$

$$\rho_i^2 = (1 - \overline{t_i}) * (\frac{\overline{t_i} * (1 - \overline{t_i})}{\sigma_{t_i}} - 1) \quad (2)$$

Where $\overline{t_i}$ is the empirical average timestamp assigned to topic i and σ_{t_i} is the empirical variance. These parameters are updated each time a word is assigned or unassigned to topic i.

3.3 Training HTMOT Using Gibbs Sampling

Two methods are commonly used for training topic models : Gibbs sampling and SVI. Gibbs sampling is asymptotically exact, i.e. it can exactly approximate the target distribution, unlike SVI (Blei et al., 2017). However, classical implementations of Gibbs sampling are prohibitively slow as they require sampling from all distributions (see algorithm 1).

Nevertheless, in the context of topic modeling, we can avoid this issue (Xiao and Stibor, 2010) and greatly speed up the process. Specifically, it is possible to only draw from the word-topic assignment distribution. This requires the construction of a data structure tailored to the model to implicitly represent the other distributions. This is the role played by our Infinite Dirichlet Trees.

As stated in section 3.1, IDTs model the aforementioned distributions based on how words are assigned to them. Hence, simply by iteratively rearranging the words in the trees, we are implicitly optimizing these distributions. This is the key to speed up the Gibbs sampling process and represents our secondary contribution.

Hence, our training procedure consists essentially of three steps (see figure 2). For each word of each document in the corpus :

1. Unassign the word from its current topic (and its ancestors) in the corpus and associated document tree.

2. Draw a topic assignment for that word from the word-topic assignment distribution.

3. Re-assign the word to the chosen topic (and its ancestors) in the corpus tree and associated document tree.

This procedure is repeated until convergence. Note that, changing a word's topic assignment will also update the estimated time parameters of the affected topics (equation 1). The initialization procedure of our algorithm is similar except that it ignores the first step as all words start unassigned.

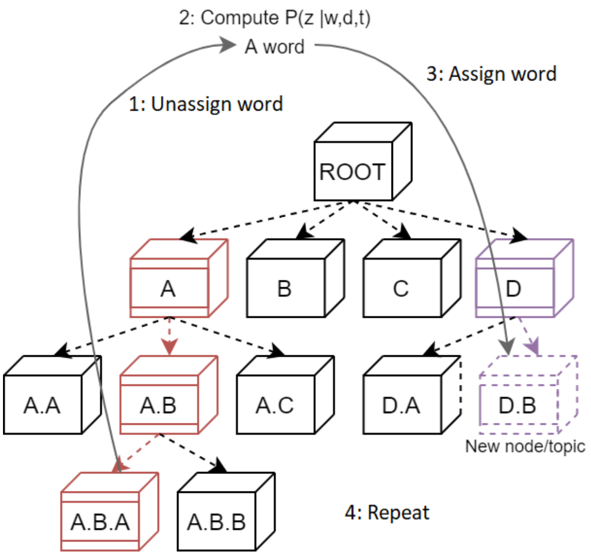

Figure 2: Gibbs sampling with Infinite Dirichlet Trees. Repeat for each word of each document until convergence.

3.3.1 Sampling Topic-Word Assignments (Paths in the Trees)

We will now explain the procedure behind sampling from the word-topic assignment distribution. When drawing a topic assignment for a word we have three possible outcomes: (1) We draw a node/topic from the associated document tree, (2) We draw a node/topic from the corpus tree or (3) We create a new node/topic.

Formally, given a word w with timestamp t in document d, we wish to draw a new topic assignment z. As stated in section 3.1, topics are identified as a sequence of node ids. Thus, we iteratively draw the random sequence $z_{0,L} = (z_0, ..., z_L)$. The length L of this sequence is decided by sampling a Bernoulli distribution in-between the sampling of each z_j.

Hence each z_j is sampled as :

$$z_j|w, d, t \sim$$

$$\begin{cases} with\ probability\ \frac{n_d}{\alpha+n_d}: & (3) \\ \sum_k \frac{\beta_k(t)*(A(k|d)+\epsilon)*(A(k|w)+\phi)*\delta_k}{(A(k)+(\phi*V))*n_d} & (4) \\ & (5) \\ with\ probability\ \frac{n_w}{\beta+n_w} * (\frac{\alpha}{\alpha+n_d}): & (6) \\ \sum_k \frac{\beta_k(t)*(A(k|w)+\phi)*\delta_k}{n_w} & (7) \\ & (8) \\ with\ probability\ \frac{\beta}{\beta+n_w} * \frac{\alpha}{\alpha+n_d}: & (9) \\ Create\ a\ new\ topic & (10) \end{cases}$$

Note that sampling a node from the corpus tree can lead to the creation of a new node in the associated document tree if that node does not already exist. However, when creating an entirely new node, it is created in both trees (corpus tree and associated document tree).

Once a topic z_j is drawn, we draw from a Bernoulli with parameter p to decide if we stop or go deeper in the tree:

$$p = \frac{P + \theta_1}{N + \theta_1 + \theta_2 + C + P} \tag{11}$$

.

$$P = \frac{\beta_j(t) * (A^*(z_{0,j}|w) + \phi) * (A^*(z_{0,j}|d) + \epsilon)}{A^*(z_{0,j}) + (\phi * V)} \tag{12}$$

$$N = \frac{\phi * \epsilon}{\phi * V} \tag{13}$$

$$C = \sum_k \frac{\beta_k(t) * (A(k|w) + \phi) * (A(k|d) + \epsilon)}{A(k) + (\phi * V)} \tag{14}$$

With $A^*(z_{0,j})$: the number of words assigned to topic $z_{0,j}$. P : the weight of the currently selected node $z_{0,j}$. C : the weight of all of the children of the selected node $z_{0,j}$. N : the weight of a potentially new child for $z_{0,j}$ and θ_1 / θ_2 : the priors for the Bernoulli distribution.

To summarize, when drawing a topic assignment for a word, we either draw from the document tree, corpus tree, or we create a new topic. Then, we draw from a Bernoulli to decide if we go deeper or not. If we do go deeper, we repeat the same process until we eventually stop. This process is then applied repeatedly too all of the words in the corpus multiple times until convergence.

3.4 Comparing HTMOT vs. nHDP

The main difference between HTMOT and nHDP is their use of Gibbs sampling and SVI training procedures, respectively. However, other notable differences exist. Firstly, our HTMOT algorithm

Variable	Description
n	# words in the corpus
n_d	# words in the corpus that are part of document d
n_w	# words in the corpus that are instantiations of the word w
V	Vocabulary length
$A(k\|w)$	# words w assigned to topic $(z_{0,j-1}, k)$ or its descendants (corpus tree information)
$A(k\|d)$	# words in document d assigned to topic $(z_{0,j-1}, k)$ or its descendants (document tree information)
$A(k)$	# words assigned to topic $(z_{0,j-1}, k)$ or its descendants
β_k	Probability density function of the beta distribution with parameter ρ_k^1 and ρ_k^2 associated with topic $(z_{0,j-1}, k)$
$\epsilon, \phi, \beta, \alpha$	Priors for the Dirichlet distributions and processes (more details are provided in the parameter section)

Table 1: Descriptions of variables for equations 3 to 10.

Variable	Description
$A^*(z_{0,j})$	Stricter version of A(*) which does not count descendant
P	Weight of the currently selected node $z_{0,j}$.
C	Weight of all of the children of the selected node $z_{0,j}$.
N	Weight of a potentially new child for $z_{0,j}$
θ_1 and θ_2	Prior for the Bernoulli distribution

Table 2: Descriptions of variables for equations 11 to 14.

starts with all words unassigned, while nHDP uses a pre-clustering step with k-means. Secondly, we do not use a greedy algorithm to select trees for each document. Instead, the tree for each document is created automatically as the Gibbs sampler progresses. As a result, our training algorithm is simpler and easier to implement, avoiding the need for pre-clustering or greedy procedures.

4 Experimental Setup

4.1 Dataset

To perform our experiments, we crawled 62k articles from the Digital Trends [1] archives from 2015 to 2020. The crawling was performed using Python with the help of the BeautifulSoup library. Digital Trends is a news website that mainly focuses on technological news but also contains general news. For all articles, we extracted the text, title, and timestamp.

The timestamps were mapped to a number between 0 and 1, which corresponds to the domain of the beta distribution used. Hence, 0 corresponds to the earliest date of a document in the corpus, and 1 corresponds to the latest.

We cleaned the data as follows. First, we removed common editor's sentences such as "*we strive to help our readers....*" to remove noise from the data. Then, we relied on Spacy's Named Entity Recognition (NER) and Part-of-Speech (POS) to filter relevant tokens [2]. Specifically, we kept specific kinds of entities (Person, Norp, Fac, Org, Gpe, Loc, Product, Event, Work_Of_Art, Law, Language) and POS elements (ADJ, NOUN, VERB, INTJ, ADV). Finally, lemmatization was also applied.

A good pre-processing procedure is essential for the interpretability of topics, as shown in (Martin and Johnson, 2015). Hence, our extraction of named entities aims to enhance the topics' interpretability by showing actors in the topic such as personalities and companies. The training algorithm will not discriminate between words and entities, but the visualization interface does. This means that a topic is no longer displayed as a simple list of words but is instead represented by a list of words and a list of entities.

4.2 Parameters

Many parameters control the behavior of our model; this section will describe each of them.

First, we have the Infinite Dirichlet Trees parameters. α : the rate at which we create new topics in the document trees. β : the rate at which we create new topics in the corpus tree. θ : how likely we are to create deeper sub-topics.

Second, we have parameters that regulate the growth of the trees. These help speed up the algorithm and keep memory usage to a minimum. CM (Critical Mass) : the minimum valid size of a

[1] https://www.digitaltrends.com/.

[2] https://spacy.io/

topic; only valid topics are part of the final output. SM (Splitting Mass) : the minimum size of a topic before it can create sub-topics. Both are defined as a percentage of the total number of words in the corpus. TTL (Time To Live) : how many pass through the corpus before destroying a non-valid node. Nodes are also destroyed when they become empty.

Third, we have the Dirichlet prior parameters as in the traditional LDA model. ϕ : the prior for the topic-word distribution. ϵ : the prior for the corpus and document-topic distributions.

Finally, we have training parameters. Iterations : how many batches we will go through during training. SGI (Stop Growth Iteration) : a point at which node new nodes won't be created. Set SGI $<$ Iterations to ensure that the last topic to be created has time to converge.

Table 3 defines the value of each parameter used to perform our experiments.

Parameter	Value
α	0.00005
β	0.0002
θ	0.25
Critical Mass (CM)	0.0005
Splitting Mass (SM)	0.005
Time To Live (TTL)	2
ϕ	0.1
ϵ	1
Iterations	4500
Batch size	500

Table 3: Parameters used for our model.

5 Results and Discussion

We now present our results, starting with a statistical analysis of the training behavior of HTMOT. Then, we will discuss the results of the Word Intrusion task, its drawbacks, and directions for future topic modeling evaluation methods. Finally, we will examine the various extracted topics qualitatively.

5.1 Convergence Tate, Training Speed, and Algorithmic Complexity

To assess the convergence of our method during training, we looked at the frequency of depth 1 topics over time. As these frequencies stabilize, it indicates that the model has converged. Since hierarchical topic models extract hundreds of topics, it is not reasonable to observe the convergence of each topic.

Our experiments revealed that the convergence rate of our training algorithm is sub-linear with respect to the dataset size. Using a dataset ten times smaller leads to a halving of the time to convergence. However, new topics created during training can perturb this convergence, which is prevented by the SGI parameter (see section 4.2).

To compare training times, we disabled HTMOT's temporal modeling to ensure a fair comparison with nHDP, which lacks a temporal component. Our sampler analyzes 135k documents per hour, while nHDP's SVI analyzes roughly 90k articles per hour, based on figures reported in (Paisley et al., 2015). Contrary to previous wisdom that SVI is considerably faster than Gibbs sampling, our training algorithm is comparable in terms of speed. The algorithmic complexity is linear with respect to the dataset size, but the depth of topic trees and growth and regulating parameters for the IDTs can greatly impact performance.

Overall, our model achieved convergence after 10 hours of training on the full dataset on commodity hardware.

5.2 Results of the Word Intrusion Task

We evaluated our model using the automated Word Intrusion task, replicating the original study(Lau et al., 2014). Unlike the classical task, we selected intruder words only from sibling topics, making the task more challenging as deeper topics tend to be more lexically related to their siblings. This is important as it helps ensure topic distinctiveness. For example, when selecting an intruder word for "astronomy", we chose from its sibling topics like "astronaut", making the chosen intruder semantically closer to the target topic. This approach provides a more robust evaluation of topic quality.

We observed an accuracy of 98% which is similar to LDA's performance (Chang et al., 2009). This demonstrates that HTMOT provides topics of similar quality with the added benefit of modeling temporality and hierarchy.

5.3 Qualitative Examination of the Resulting Topics

In figure 4, our model's ability to extract atomic events at the deeper level of the tree is demonstrated through the well-localized time distribution of the three sub-topics under "astronauts". These sub-topics, namely the historic test launch of the

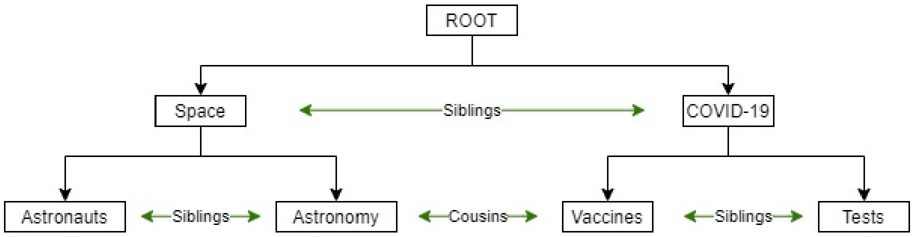

Figure 3: Example of a topic tree with cousins and siblings.

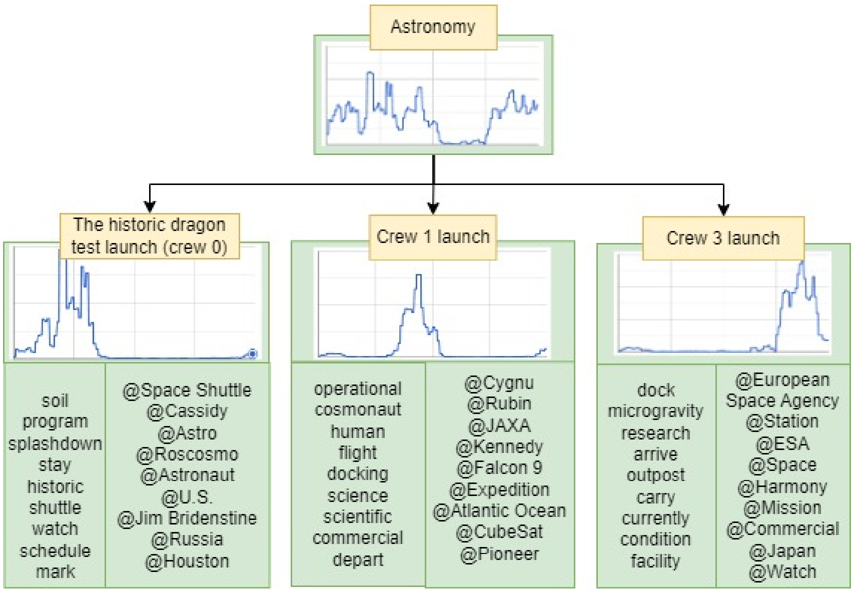

Figure 4: Examples of depth 3 topics that are well localized in time.

spaceX Dragon capsule, the crew 1 launch, and the crew 3 launch, were mostly interpreted from top documents due to their depth, making it difficult to interpret based on top words. The timing of these events matched their associated time distribution, occurring in May 2020, November 2020, and November 2021 respectively. The model missed the crew 2 launch event, which may be related to the reduced output of digital trends news during that period, as shown in figure 5.

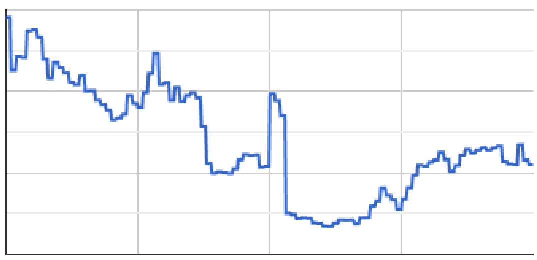

Figure 5: Number of articles published by Digital Trends over the years 2020 and 2021. We can see a sharp decline at the beginning of the year 2021 (middle of the graph).

6 Conclusion

We have proposed a new model for topic modeling capable of modeling hierarchy and time jointly. Through examples, we have demonstrated how combining hierarchy and temporality provides us with a more fine-grained understanding of a corpus through detailed sub-topics which can represent specific events. Moreover, we developed a novel implementation of Gibbs sampling for hierarchical topic models. This implementation provides a fast alternative to SVI that makes Gibbs sampling a viable solution for training such complex models. Moreover, we have shown how extracting entities can help interpret and understand topics at a deeper level.

References

Rubayyi Alghamdi and Khalid Alfalqi. 2015. A survey of topic modeling in text mining. *International Journal of Advanced Computer Science and Applications*, 6(1).

Bhagyashree Vyankatrao Barde and Anant Madhavrao

Bainwad. 2017. An overview of topic modeling methods and tools. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE.

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. 2017. An automatic approach for document-level topic model evaluation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 206–215, Vancouver, Canada. Association for Computational Linguistics.

David M Blei, Thomas L Griffiths, Michael I Jordan, and Joshua B Tenenbaum. 2004. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16(16):17–24.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, volume 22, pages 288–296. Curran Associates, Inc.

Caitlin Doogan and Wray Buntine. 2021. Topic model or topic twaddle? re-evaluating semantic interpretability measures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3824–3848, Online. Association for Computational Linguistics.

Manal El Akrouchi, Houda Benbrahim, and Ismail Kassou. 2021. End-to-end lda-based automatic weak signal detection in web news. *Knowledge-Based Systems*, 212:106650.

Liangjie Hong, Dawei Yin, Jian Guo, and Brian D. Davison. 2011. Tracking trends: Incorporating term volume into temporal topic models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 484–492, New York, NY, USA. Association for Computing Machinery.

Alexander Miserlis Hoyle, Pranav Goel, Denis Peskov, Andrew Hian-Cheong, Jordan L. Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken?: The incoherence of coherence. *CoRR*, abs/2107.02173.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden. Association for Computational Linguistics.

Fiona Martin and Mark Johnson. 2015. More efficient topic modelling through a noun only approach. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 111–115, Parramatta, Australia.

David Mimno, Wei Li, and Andrew McCallum. 2007. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, pages 633–640.

Ramesh M Nallapati, Susan Ditmore, John D Lafferty, and Kin Ung. 2007. Multiscale topic tomography. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 520–529.

J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. 2015. Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270.

Jay Pujara and Peter Skomoroch. 2012. Large-scale hierarchical topic models. In *NIPS Workshop on Big Learning*, volume 128.

Yang Song, Lu Zhang, and C Lee Giles. 2008. A non-parametric approach to pair-wise dynamic topic correlation detection. In *2008 Eighth IEEE International Conference on Data Mining*, pages 1031–1036. IEEE.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Hei-Chia Wang, Tzu-Ting Hsu, and Yunita Sari. 2019. Personal research idea recommendation using research trends and a hierarchical topic model. *Scientometrics*, 121(3):1385–1406.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 424–433, New York, NY, USA. Association for Computing Machinery.

Han Xiao and Thomas Stibor. 2010. Efficient collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of 2nd Asian Conference on Machine Learning*, volume 13 of *Proceedings of Machine Learning Research*, pages 63–78, Tokyo, Japan. JMLR Workshop and Conference Proceedings.

Xiangmin Zhou and Lei Chen. 2013. Event detection over twitter social media streams. *The VLDB Journal*, 23(3):381–400.

Xiaofeng Zhu, Diego Klabjan, and Patrick N. Bless. 2017. Unsupervised terminological ontology learning based on hierarchical topic modeling. In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 32–41.