HTMOT : Hierarchical Topic Modelling Over Time

Judicael POUMAY[a], Ashwin ITTOO[a]

[a]*ULiege/HEC Liege Rue louvrex 14, 4000 Liege, Belgium*

Abstract

Over the years, topic models have provided an efficient way of extracting insights from text. However, while many models have been proposed, none are able to model topic temporality and hierarchy jointly. Modelling time provide more precise topics by separating lexically close but temporally distinct topics while modelling hierarchy provides a more detailed view of the content of a document corpus. In this study, we therefore propose a novel method, HTMOT, to perform Hierarchical Topic Modelling Over Time. We train HTMOT using a new implementation of Gibbs sampling, which is more efficient. Specifically, we show that only applying time modelling to deep sub-topics provides a way to extract specific stories or events while high level topics extract larger themes in the corpus. Our results show that our training procedure is fast and can extract accurate high-level topics and temporally precise sub-topics. We measured our model's performance using the Word Intrusion task and outlined some limitations of this evaluation method, especially for hierarchical models. As a case study, we focused on the various developments in the space industry in 2020.

Keywords: topic modelling, temporality, hierarchy, Gibbs sampling, topic interpretability

1. Introduction

The amount of data being generated by humanity increases exponentially. About 80% of the unstructured data worldwide is textual and comes from a wide variety of ubiquitous sources, including online news and social media platforms. Buried within those voluminous amounts of texts are meaningful insights, which could help in supporting business decision-making activities. Hence, methods for extracting these insights are becoming more and more valuable.

Thematic information constitutes one such type of insight. In NLP, several methods for topic detection from text have been proposed to extract the various themes contained in a corpus[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. These models have been employed for different applications, including fraud detection [11], environment scanning [12], understanding employee and customer satisfaction [13, 14] among others [15, 16, 17].

By far, Latent Dirichlet Allocation (LDA) [1] is the most commonly employed topic model. However, despite its widespread adoption, it suffers from a number of longstanding limitations. First, LDA requires the number of topics to be defined upfront. This constraint arises from the underlying Dirichlet law used, which defines a distribution over random vectors of finite length. The finite nature of these vectors forces users to pre-define a number of topics to extract.

Email addresses: judicael.poumay@uliege.be (Judicael POUMAY), ashwin.ittoo@uliege.be (Ashwin ITTOO)

Consequently, users are compelled to experiment with a varying number of topics in order to determine the optimal number for their applications. Additionally, LDA ignores interesting characteristics of topics, in particular, hierarchy and temporality.

To overcome the issue of predefining the number of topics, the Hierarchical Dirichlet Process model (HDP) [2] has been proposed. Through the use of Dirichlet Processes, it is capable of deciding the number of topics during training. This is because contrary to Dirichlet laws, a Dirichlet Process defines a distribution over random vectors of infinite length. In practice this provides the ability to determine the number of topics dynamically during training. HDP has been used in many tasks such as speaker diarization [18], i.e. clustering an input audio stream according to the number of speakers, determined automatically.

More importantly, HDP provided a solid premise for the development of more advanced topic modelling methods, such as the nested Chinese Restaurant Process (nCRP) [3] and the nested Hierarchical Dirichlet Process (nHDP) [4]. The latter being an improved version of the former. nHDP was tested on news and Wikipedia articles and evaluated based on a perplexity and a qualitative analysis; both analyses revealing a substantial improvement over LDA and HDP. Moreover, nHDP was trained using Stochastic Variational Inference (SVI) which is substantially faster than the traditional Gibbs Sampling algorithm used. Such hierarchical models enable the extraction of topics and corresponding subtopics, organizing them in a tree-like hierarchy of arbitrary depth and breadth. Similar to HDP, the shape of this tree is defined during training thanks to the use of Dirichlet processes. Extracting topic hierarchies from a corpus provides a more fine-grained view of the underlying data and is particularly useful in applications such as ontology learning [19] and research idea recommendation[20].

In parallel, temporal topic models [6, 8, 9, 7] have been proposed to model the evolution and/or localization of topics with respect to time. For instance, in the seminal study of Wang and McCallum [6], the authors associated each topic with a Beta distribution to model time. They showed that their Topic over Time (ToT) model more accurately pinpointed the occurrence of topics in time, augmenting the results obtained by classical LDA. A KL-divergence analysis further revealed that the topics extracted by their model were more distinct than those extracted by LDA. In essence, incorporating temporal information enable the separation of topics that are lexically similar but temporally distinct. ToT was evaluated on various datasets, such as NIPS articles, e-mails and historical texts. Temporal topic models have been used for tracking trends in scientific articles [21] and events in social media [22].

Intuitively, incorporating both temporal and hierarchical modelling would yield models that encompass the strengths of both. Furthermore, several applications warrant the incorporation of temporal and hierarchical information in topic models. One such application is that of environment scanning [12], which is the task of gathering, analyzing and monitoring information that is relevant to an organization to identify future threats and opportunities. It is clear that this task would benefit from having both more detailed topics using hierarchical modelling and more precise topics that are better

localized in time using temporal modelling. However, to date, no topics model integrating both temporal and hierarchical information exist, despite the recent advances in topic modelling and in NLP in general.

To address this issue, we propose a novel method, HTMOT, for Hierarchical Topic Modelling Over Time. By jointly modelling topic hierarchy and temporality, our model offers the advantages of previous methods, which only focused on a single dimension (i.e. temporality or hierarchy). To the best of our knowledge, our model is the first to jointly model topic hierarchy and temporality.

However, incorporating both temporal and hierarchical information poses a significant challenge during training. While HTMOT is based on nHDP, it could not be trained using Stochastic Variational Inference (SVI) like its predecessor. This is because SVI requires all distribution to estimate to have a conjugate prior. However, the beta distribution used to model time in ToT does not have a known conjugate prior and therefore SVI could not be used to train HTMOT. To overcome this issue, we resorted to using Gibbs sampling which is known to be slow and would result in an prohibitively long training time.

To address this problem, we developed a new implementation of Gibbs sampling for HTMOT. At its core is a novel tree-based data structure, which we call the *Infinite Dirichlet Tree*. The basic idea is to assign all the words in the corpus to the nodes of the tree which represents topics. More specifically, there is one such tree for the corpus and one for each document. As we will see later, the arrangement of words in the trees implicitly defines a topic-word, topic-time, document-topic, corpus-topic and topic-hierarchy distributions. The Gibbs sampler is then reduced to the sampling of only one distribution (the word-topic distribution) which decides to which nodes each word is assigned to in an iterative manner. This implicitly estimates the others distributions as the sampler constantly re-arranges the words in the tree. Consequently, this implementation of Gibbs sampling provides a massive speed-up compared to traditional implementation. Specifically, our experiments reveal that it is similar to SVI in terms of speed. Our HTMOT model and its novel Gibbs sampling training procedure constitute the core contributions of our work.

As ancillary contributions, we demonstrate how NER could help enhancing the interpretability of topics. Additionally, the theoretical information about Dirichlet Processes necessary to develop this implementation was scattered around the literature. Hence, we also provide a synthetic review of Dirichlet Processes for hierarchical topic modelling.

Finally, we evaluate our method's performance using various methods, including the recent Word Intrusion task [23]. It involves human users/annotators detecting an intruder (polluting) word, deliberately introduced in a topic. Given the difficulty in evaluating topics [23], we provide a critic of the Word Intrusion task, in particular in the context of hierarchical topic modelling, and propose directions for future topic evaluation methods.

Our literature review showed that there is no standard dataset for evaluating topic models. Therefore, for our experiments we crawled 62k news articles from DigitalTrends.com between 2015 and 2020. Doing so enabled us to test our method on recent news. As a case study, we focused on the various developments in the space industry in 2020 to show

how our HTMOT method can help uncover insights about a particular domain to perform environment scanning. Indeed, the results do provide a better understanding of the many recent developments in the space industry such as the advent of new spacecrafts (Boeing Starliner, NASA Orion), and the heightened interest in asteroid sampling missions.

For the Word Intrusion task, 57 people answered our survey which resulted in a 74.83% overall accuracy. More precisely, we show that deeper topics show worst performance: 98.25% for depth 1 topics and 63.13% for depth 2. However, we also measured the confidence of annotators: 92.63% for depth 1 topics while depth 2 topics show 71.13% confidence. Hence, we argue that deeper topics require more knowledge from annotators which is one of the flaw of the Word Intrusion task for hierarchical models.

To summarize, our contributions are:

1. A novel method, HTMOT, for hierarchical topic modelling over time
2. A fast Gibbs Sampling implementation based on a novel tree-based data structure
3. Increasing the interpretability of topics through NER extraction
4. Providing a critic of the Word Intrusion task
5. Providing a synthetic review of Dirichlet Processes for hierarchical topic modelling

The rest of the paper is structured as follow. Section 2 provides an overview of Dirichlet processes and describes current topic modelling studies. Then, in section 3 we present our model by describing *Infinite Dirichlet Trees* , how temporality was incorporated, and the training procedure. Next, section 4 will present the experimental setup, the evaluation methods, and parameters used. Finally, results are presented in section 5.

2. Background and Related Work

In this section, we begin with a synthetic review of Dirichlet processes. Then, we will review the literature on topic models that are related to ours. Finally, we will discuss the various methods used to evaluate topics.

2.1. Dirichlet processes (DPs)

To understand the foundation of our model, it is important to understand Dirichlet Processes. However, the important information about Dirichlet Processes, especially in the context of topic modelling is scattered across different sources in the existing literature. Hence, we provide a synthetic review of the DP from its definition to its important properties with respect to topic modelling. We believe that our synthesis could be valuable for future related research.

A Dirichlet distribution defines a probability distribution over random vectors of finite dimensions defined such that the sum of their elements is one; Dirichlet processes generalize this idea to infinity. Specifically, a Dirichlet process defines a probability distribution over random vectors of infinite dimensions (for which the sum of their elements is one).

Formally, let H be a probability distribution (called base distribution) over S and α a positive real number. Then, we can define a Dirichlet Process, $DP(H, \alpha)$, as a stochastic process whose realization X is itself a probability distribution over S, such that for any measurable finite partition of S, denoted $\{B_i\}_{i=1}^{n}$:[1]

$$X \sim DP(H, \alpha) \Rightarrow (X(B_1), ..., X(B_n)) \sim Dir(\alpha H(B_1), ..., \alpha H(B_n))$$

Using the stick breaking construction (see figure 1) we can define X as :[2]

$$X = \sum_{i}^{\infty} v_i \delta_{\theta_i}$$

- $\theta_i \sim H$; $\theta_i \in S$
- $v_i = \pi_i \sum_{k=0}^{i-1}(1 - \pi_k)$; $\pi_i \sim Beta(1, \alpha)$

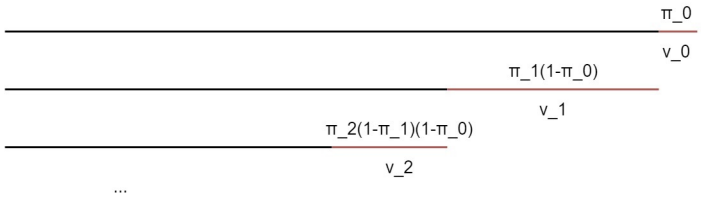


Figure 1: Visualisation of the stick breaking construction

H can be any distribution but in the context of topic modelling, H is a Dirichlet Distribution. Then, each θ_i corresponds to a topic-word distributions, v is the infinite random vector drawn from the Dirichlet process and each v_i corresponds to the probability of each topic. The stick breaking construction defines how the probabilities are distributed over the space of topics S. This construction implies that a small proportion of topics are assigned most of the probability mass. The spread of the probability mass across the topics is controlled by the concentration parameter α.

DPs have three properties that are interesting for hierarchical topic modelling: discreteness, clustering, and hierarchisation. **Discreteness** : We can look at a DP as a tool to discretize a distribution. Independent of whether or not H is continuous, the resulting $X \sim DP(H, \alpha)$ is defined as a weighted sum of discrete atoms. This discreteness ensures that the probability of a topic is not infinitesimal and leads to the clustering property. **Clustering** : Let $\theta_1, ..., \theta_n$ be a sample of topics drawn from X. Then, if we were to draw from the posterior $X|\theta_1, ..., \theta_n$, we would have a reasonable chance

[1] Dir(.) correspond to the Dirichlet law
[2] With δ_x being the Dirac delta distribution.

of drawing from $\theta_1, ..., \theta_n$ again. Precisely, the posterior is :

$$X|\theta_1, ..., \theta_n \sim DP(\frac{\alpha}{\alpha+n} * H + \frac{n}{\alpha+n} * \sum_i^n \delta_{\theta_i}, \alpha + n)$$

Thus, as the number of observations (n) increases, we start to ignore H and focus on the empirical distribution $\sum_i^n \delta_{\theta_i}$. Hence, we are more likely to choose atoms that were already chosen, and clustering happens. However, we always retain some probability of drawing from H and create a new cluster (topic). In our case, n is equal to the number of words in the corpus. **Hierarchisation** : If we consider another Dirichlet Process $Y \sim DP(X, \beta)$ with its own concentration parameter β, then the atoms of Y are exactly the atoms of $X \sim DP(H, \alpha)$. This property can be used so that documents and corpus share the same topics (atoms) but with a different distribution over them. Precisely, X will model the corpus-topic distribution while multiple Y_i can be used to model each document-topic distribution. For more information about DPs, see Ghosal [24], Teh [25] and Paisley et al. [4].

2.2. Topic Modelling

We now describe related studies on topic modelling methods. However, given the significant attention that topic modelling has received over the years, we will not provide a comprehensive review, which is out of the scope of our study. Instead, for conciseness, we will focus only on those studies most closely related to ours. For more in-depth reviews see Alghamdi and Alfalqi [26] and Barde and Bainwad [27]. Also, to better present the wide diversity in datasets and evaluation methods, we list them in table 1.

Topic Modelling. LDA [1] model was the first popular topic model. It has provided the basis for subsequent models and is still at the crux of many applications. At the core of LDA is a Bayesian generative model, with two Dirichlet distributions, respectively for the document-topic distribution and for the topic-word distribution. These distributions are learnt and optimized via an inference procedure, which enables topics to be detected. 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 by optimizing some performance metric, such as the perplexity metric.

A more systematic approach to to determine the number of topics is to rely on DPs instead of Dirichlet distributions, as done with HDP Teh et al. [2] . As we discussed earlier, the difference between Dirichlet distributions and processes is that DPs define a distribution over random vectors of infinite length. Hence, for DPs there is no limit to the number of topics. Yet the discrete and finite nature of a corpus forces the collapse of the potentially infinite number of topics into a finite set of clusters of words. Thus, the final number of topic is finite but determined automatically during training. Otherwise, HDP operates on a similar premise to LDA. At the corpus and document level, we have distributions over the topics (DPs) and at the topic level, we have a distribution over the words (Dirichlet distribution). The hierarchisation

property of the DPs ensures that the atoms (topics) of the global (corpus) DP are shared by the local (document) DPs; they define different distributions over the same topics.

Topic models such as LDA have seen many application including fraud detection [11], environment scanning [12], understanding employee and customer satisfaction [13, 14] and many other applications [15, 16, 17]. In particular, El Akrouchi et al. [12] used LDA to extract potential weak signals from news articles.

Hierarchical Topic Modelling. Methods such as LDA and HDP provide interesting insights into the content of a corpus. However, these methods are only capable of extracting the higher order topics. Hence, new methods have been developed to also extract topic hierarchies which provide more fine-grained insights into the content of a corpus.

One method for hierarchical topic modelling is nHDP [4]. It models topic hierarchy by defining a potentially infinite tree where each node corresponds to a topic. At each level 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, then we will choose a sub-topic in the same way we choose the parent topic except that now we work with a sub-corpus made up of the tokens assigned to the parent topic.

Other topic models have been proposed to model hierarchy [10, 5, 3]. hPAM [10] proposes an acyclical graph structure instead of a tree to model topic hierarchy where high level topics can share low level topics while also modelling topic correlation; while this provides more precise relationships between topics, it is harder to display and navigate. LSHTM [5] recursively applies LDA to the sub-corpus defined by the topics of the previous LDA application; hence, it requires a pre-defined set of parameters to define the shape of the final topic tree. Finally, the nCRP [3] is the predecessor of nHDP and works similarly except that it does not model the document-topic distribution resulting in mono-topic document modelling. Hence, nHDP is more powerful than [5, 3] while keeping a strict tree structure contrary to [10] which is easier to display.

Extracting topic hierarchies provides a more fine-grained view of the data and is particularly useful in applications such as ontology learning [19] and research idea recommendation[20]. In particular, Wang et al. [20] used Hierarchical Topic Modelling to discover a user's interest from articles read and cross-referenced these interests with current research trends to provide research ideas.

Temporal Topic Modelling. Other authors have decided to investigate the temporality of topics. This provides information such as when a topic occurred or how it evolved. Understanding the temporality of topics is important, especially for environment scanning where changes in the environment are the most important signals.

ToT [6] proposes a modified version of LDA by incorporating 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 [9, 7, 8]. MTT [8] creates a binary tree which provides the ability to understand topics at various time scale; deeper nodes correspond to a smaller timescale. DTM [7] 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, DCTM [9] also slices the corpus in period but uses Gaussian processes and SVD instead of LDA based techniques. The advantage of ToT is that it is non-Markovian and it models time as a continuum. Continuity, in particular, is important when mixing ToT with nHDP as we are already building a tree for the topic hierarchy and ToT is the only model which does not require its own discrete structure to model time such as slices or a binary tree.

Such Temporal models have been used for tracking trends in scientific articles [21] and events in social media [22]. In particular, Zhou and Chen [22] uses a ToT-based topic model that also incorporates geoposition information to extract events from social media.

2.3. Topic Models Evaluation

Various methods have been used in previous studies to evaluate topic models as we can see in table 1.

Perplexity has been the standard for comparing topic models for a long time. It requires to compute how likely it is that the data would have been generated by the trained topic model. However, this method does not correlate with human judgement [23]. Hence, new methods for evaluating topics have been proposed, but none have provided a new standard.

A novel evaluation method is the Word Intrusion task. It involves inserting an intruder word in the list of topic words, then ask people to find this intruder [23]. The idea is that in good topics, the annotators would easily find it. This intruder is selected at random from a pool of words with low probability in the current topic but high probability in some other topic to avoid rare words. With this evaluation method, the final score corresponds to the average classification accuracy made by humans. In, [28] they have shown that this task can be automated with performance close to human annotators.

Newman et al. [29] proposed topic coherence as a method of topic evaluation. This method consist in computing some similarity score between the top N topic words. Precisely, it is computed as (where w_i is more frequent than w_j): $\sum_{i<j} score(w_i, w_j)$. Topic coherence is a modular evaluation method as it allows for many different score functions; UCI and UMass being the most popular. Both use the co-occurrence of words but UCI is an extrinsic measure based on Wikipedia articles while UMass is intrinsic and uses the training corpus. However, other score functions such as the cosine similarity of word embeddings can also be used. The topic coherence score of a model is the average coherence score of the topics. In this article we will use UMass as an intrinsic evaluation method and Word Intrusion as an extrinsic one.

Finally, all papers listed in table 1 provide a qualitative analysis of the extracted topics. Compared to opaque measures such as coherence and perplexity, qualitatively examining the resulting topics provides a good understanding of the model's performance. However, such an evaluation method is prone to cherry picking especially when many topics are extracted.

Model name	Corpora	Evaluation	Type
LDA [1]	TREC (news articles) Biomedical abstracts Reuters gold dataset	Document modelling (Perplexity) Classification (Accuracy) Collbaborative filtering (Perplexity) Qualitative	Classic
HDP [2]	NIPS articles	Document modelling (Perplexity) Qualitative	Classic
nCRP [3]	JACM abstracts	Document modelling (Perplexity) Qualitative	Hierachical
PAMmix [10]	Medicine abstracts	Document modelling (Perplexity) Classification (Accuracy) Qualitative	Hierachical
nHDP [4]	New York time Wikipedia articles	Document modelling(Perplexity) Qualitative	Hierachical
LSHTM [5]	TREC (news articles) Wikipedia	Word Intrusion (Accuracy) Qualitative	Hierachical
DTM [7]	Science articles	Document modelling (Perplexity) Qualitative	Temporal
ToT [6]	NIPS E-Mails Historical text	KL-divergence Classification (Accuracy) Qualitative	Temporal
MTT [8]	Science Magazine articles	Qualitative	Temporal
DCTM [9]	Scientific paper	Document modelling (Perplexity) Qualitative	Temporal

Table 1: Related methods corpora, evaluation methods used and type of topic model.

3. HTMOT : Hierarchical topic modelling over time

We now describe our method for hierarchical topic modeling over time, HTMOT. We begin by presenting a new type of data structure at the core of HTMOT. We refer to this tree-based structure as an *Infinite Dirichlet Tree (IDT)*. IDT's main purpose is to record the word-topic assignments. In addition, as will be detailed in section 3.3, they are at the basis of our fast implementation of the Gibbs sampling procedure. Next, we describe how temporality was modelled, before finally detailing our novel implementation of Gibbs sampling for HTMOT.

3.1. Counting words using Infinite Dirichlet Trees

An Infinite Dirichlet Tree (IDT) is an efficient tree-based data structure we developed for HTMOT. There is one such tree for the corpus and one for each document. Each node in these trees represents a topic and each word in the corpus is assigned to a node (see figure 2).They are called IDTs to refer to the potentially infinite number of topics provided by the Dirichlet Processes which defines how they grow. Every node of the infinite tree is potentially accessible, they are created when a first word is assigned to them and destroyed automatically when unused. Each node is identified by a finite path in the tree as a sequence of node ids, starting from the root. In the next paragraphs, we will see how these trees are used to model the topic-word, topic-time, document-topic, corpus-topic and topic-hierarchy distributions jointly.

We have a tree for the corpus and a tree for each document. All words in the corpus are assigned to nodes of the corpus tree while document trees only contains words from their respective document. Hence, the document trees are mutually exclusive subsets of the corpus tree in the sense that adding the content of their nodes would yield the corpus tree. For both kind of trees, each node (topics) will be assigned a different number of words meaning they will differ in size which creates a distribution. Hence, the corpus tree defines a corpus-topic distribution and each document trees define a document-topic distribution.

Each node represents a topic and each word in the corpus is assigned to a node in the corpus tree and associated document tree (this assignment is later explained in section 3.3.1). More precisely, each topic is represented by at most one node in each tree; a topic may not be represented in a document tree if no word of the associated document has been assigned to that topic. Additionally, each word is associated with the timestamp of its document. The timestamp corresponds to the date of the document and is mapped to a number between 0 and 1[3] such that 0 corresponds to the earliest date of a document in the corpus and 1 corresponds to the latest. Thus, for each node, we can count how many and which words have been assigned to it. This defines a word distribution. Moreover, since each word is associated with a timestamp, we also have a time distribution. Thus, each node defines a topic-word and a topic-time distribution.

The trees also model the hierarchical distribution of topics. When a word is assigned to a node, it is also assigned to all ancestors of that node. Hence, each word is assigned to a sequence of nodes starting from the root up to the final chosen topic. This creates a hierarchical dependency between the nodes and thus a hierarchical distribution. Hence, there are two types of assignments. First, when a word is assigned to a node as it was the chosen topic for that word; we say the word stops at that node. Second, when a word is assigned to all of the ancestor nodes of the chosen topic (node); we say the word pass through those nodes. This will be important later when discussing word assignments (section 3.3.1).

To speed up training, we also make use of multiple parameters. A Critical Mass threshold (CM) is used to define a minimal size for topic nodes. Given that N is the number of words in the entire corpus, if a node/topic has fewer than $CM * N$ assignments, it won't be considered valid. If a node is not valid, the topic it represents will not be included in the results, and if it stays below the critical mass after seeing the data twice [4], it will be destroyed automatically. Nodes are also destroyed when they become empty. In any case, words that the now destroyed node may have recorded are simply unassigned; they will be re-assigned in a later iteration. Furthermore, a similar splitting mass threshold is used to decide when nodes are large enough to create children. Finally, nodes create children one at a time and only if all the other current children are valid. These rules constrain the growth of the trees and improve run-time and memory usage.

[3]The domain of the beta distribution used

[4]This parameter is called the Time To Live (TTL) and was set after empirical observations

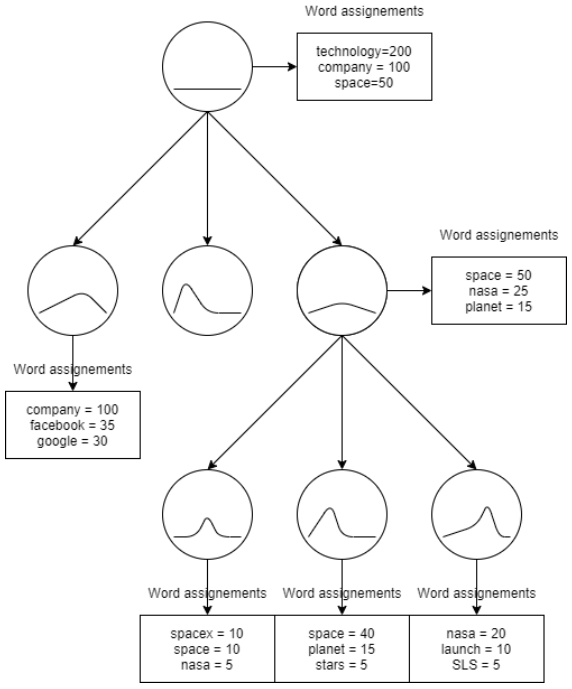


Figure 2: Example of an IDT with word assignments and time distribution (inside nodes).

3.2. Modelling temporality

Temporality is modelled by associating a topic with a beta distribution as in ToT [6]. However, contrary to ToT, we do not apply temporality to all topics but only deep ones; we choose depths 3 and 4 for our experiments. The reason is that the beta distribution is monomodal (when both of its parameters >1) which is not enough to model the complex temporality of high-level topics.

Indeed, if we consider the topic of space exploration, many events may occur throughout a given year (especially in recent ones). These multiple occurrences will translate to many peaks in frequency. Therefore, such high-level topics fundamentally have a multimodal distribution. Hence, if we were to apply time modelling to high-level topics, it would separate them into many similar monomodals topics which is undesirable. Conversely, deeper topics (i.e. subtopics) often correspond to precise stories/events, occurring within narrower time frames. Thus, they can be characterized by monomodal distributions, such as the beta distribution.

Hence, by mixing temporal and hierarchical modelling and only applying time modelling to deep topics we are able to separate temporally distinct topics at the deepest level while keeping the number of high level topics low enough that they are easily interpretable.

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). Currently, there exist no known conjugate prior for the beta distribution that is suitable for our application. Hence, we use the method of the moment to estimate these parameters[6] :

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

Where $\overline{t_i}$ is the empirical average timestamp assigned to topic i and σ_{t_i} is the empirical variance. Note that the variance is set to have a minimum of 0.0001 to avoid numerical instability and high ρ_i values. These parameters are updated each time a word is assigned or unassigned to a topic. Note that non-valid nodes are forced to have a uniform time probability to let them grow unaffected by time distribution until they reach critical mass.

Furthermore, to force topics to be more localized in time we modify the estimated parameters. More specifically, when using the beta distribution of a topic in practice, we multiply its parameters ρ_i by some constant Δ; detailed values are provided in the parameter section. Then, we add one to the resulting parameters because parameter values between 0 and 1 lead to a bi-modal regime for the Beta distribution which is not desirable. Finally, we add 0.5 to the Beta Distribution itself. This is because the beta distribution can be valued at near zero for large parameters but this is not desirable as it can lock a topic in time during training since a probability of 0 means no words would be assigned outside of its current time range.

Thus, we define a modified version of the beta distribution:

$$ModBetaPDF(\rho^1, \rho^2, \Delta) = \begin{cases} (0.5 + BetaPDF(1 + \rho^1/\Delta, 1 + \rho^2/\Delta))/1.5 & \text{if } \Delta <= 1 \quad (1) \\ 1 & \text{else} \quad (2) \end{cases}$$

Note that ModBetaPDF returns 1 for $\Delta > 1$. This is used to disable time for higher level topics as Δ values are defined for each depth.

3.3. Training HTMOT using Gibbs sampling

Our model (HTMOT) is based in part on nHDP [4] which was trained using stochastic variational inference (SVI). However, this training procedure requires each distribution to estimate to have a conjugate prior. The problem is that the beta distribution used to model time does not have a known conjugate prior. Thus, we had to use Gibbs sampling as a substitute training procedure for our model.

Gibbs sampling is advantageous as it eliminates the need for conjugate priors. Moreover, contrary to variational inference it is asymptotically exact and more precise with small datasets [30]. This is important because we care about small sub-topics which are contained in small subsets of the dataset. Finally, Gibbs sampling avoids the problem of mode collapse that is inherent to variational inference. However, its main weakness is that classical implementations are typically prohibitively slow.

Consequently, we had to develop a novel Gibbs sampling implementation to compete with SVI in terms of speed.

Algorithm 1 Traditional Gibbs sampling

procedure GIBBS($corpus$)
  for N iterations **do**
    for each $document$ in $corpus$ **do**
      for each $word$ in $document$ **do**
        Sample word-topic assignment full conditional $P(z|w, d, t, B, D, T, C, H)$
        Sample topic-word full conditional $P(B|w, d, t, z, D, T, C, H)$
        Sample document-topic full conditional $P(D|w, d, t, B, z, T, C, H)$
        Estimate time-topic full conditional $P(T|w, d, t, B, D, z, C, H)$
        Sample corpus-topic full conditional $P(C|w, d, t, B, D, T, z, H)$
        Sample hierarchy-topic full conditional $P(H|w, d, t, B, D, T, C, z)$
      end for
    end for
  end for
  Return solution : (z,B,D,T,C,H)
end procedure

While classical Gibbs sampling implementation requires sampling from all distributions (see algorithm 1), in the context of topic modelling, it is possible to only draw the word-topic assignment distribution [31] which greatly speed up the process. However, this requires the construction of a data structure tailored to the model to implicitly model the other distributions. This is the role played by our novel Infinite Dirichlet Trees.

As can be seen in figure 3), our algorithm consists essentially of three steps. Firstly, unassigning the word from its current topic (and its ancestors) in the corpus and associated document tree. Secondly, draw a topic assignment given the word, document and timestamp based on the corpus and document tree. Thirdly, re-assign the word to the chosen topic (and its ancestors) in the corpus and associated document tree. The initialization procedure of our algorithm is similar expect that it ignores the first step as all words starts unassigned.

As stated in section 3.1, the Infinite Dirichlet Trees model the topic-word, topic-time, document-topic, corpus-topic and topic-hierarchy distribution from their content. Hence, simply by iteratively re-arranging the words through the sampling of the topic assignment distribution, we are implicitly optimizing the aforementioned distributions modelled by the trees (topic-word, topic-time, document-topic, corpus-topic and topic-hierarchy). This is the key to speed up the Gibbs sampling process.

3.3.1. Sampling topic-word assignments (paths in the trees)

Given that the corpus contains *n* words, n_d of those are part of document d and n_w of those are instantiations of the word w. Then, when drawing a topic assignment z_j for a word w with timestamp t in document d at a topic depth j we may draw from three possible distributions. These correspond to 1) drawing a node from the document tree (Local DP) 2) drawing a node from the corpus tree (Global DP) 3) create a new node:

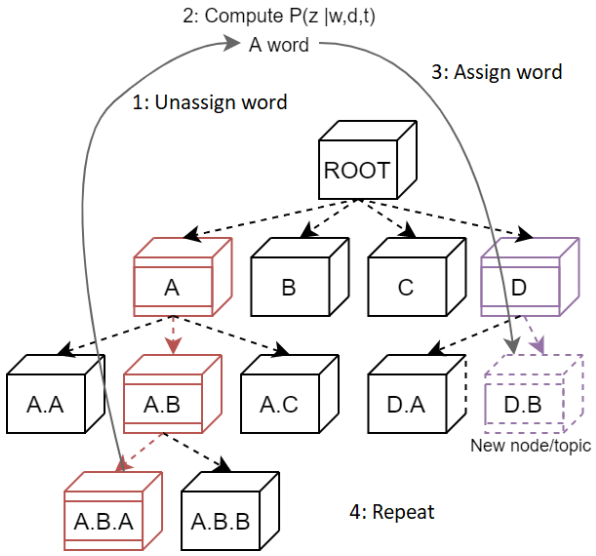


Figure 3: Gibbs sampling with Infinite Dirichlet Trees. Repeat for each word of each document until convergence.

$$z_j|w,d,t \sim \begin{cases} \sum_k \frac{ModBetaPDF(\rho_k^1,\rho_k^2,\Delta_j)(t)*(A(k|d)+\epsilon)*(A(k|w)+\phi)*\delta_k}{(A(k)+(\phi*V))*n_d} & \text{with probability } \frac{n_d}{\alpha+n_d} \quad (3) \\ \sum_k \frac{ModBetaPDF(\rho_k^1,\rho_k^2,\Delta_j)(t)*(A(k|w)+\phi)*\delta_k}{n_w} & \text{with probability } \frac{n_w}{\beta+n_w} * (\frac{\alpha}{\alpha+n_d}) \quad (4) \\ new & \text{with probability } \frac{\beta}{\beta+n_w} * \frac{\alpha}{\alpha+n_d} \quad (5) \end{cases}$$

Where, $A(k|w)$ corresponds to the number of words w assigned to topic k or its descendants (corpus tree information). $A(k|d)$ corresponds to the number of words in document d assigned to topic k or its descendants (document tree information). [5] ModBetaPDF corresponds to the probability density function of the modified beta distribution. Finally, the parameters $\epsilon, \phi, \beta, \alpha$ are priors for the Dirichlet distributions and processes; more details are provided in the parameter section.

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

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

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

.

$$P = \frac{ModBetaPDF(\rho_p^1, \rho_p^2, \Delta_j)(t) * (A^*(parent|w) + \phi) * (A^*(parent|d) + \epsilon)}{A^*(parent) + (\phi * V)}$$

[5] Remember that k identifies a node as a path (sequence of ids) starting at the root.

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

$$C = \sum_i \frac{ModBetaPDF(\rho_i^1, \rho_i^2, \Delta_j)(t) * (A(child_i|w) + \phi) * (A(child_i|d) + \epsilon)}{A(child_i) + (\phi * V)}$$

Where, P is the weight of the currently selected parent node. C is the weight of all of the children and N is the weight of a potentially new child. These weights are updated each time a word instance is (re)-assigned to a node. V is the vocabulary length and $A^*(k)$ is a strict version of $A(k)$ which does not count elements that pass through the node k only those which stops. Finally, θ_1 and θ_2 are the prior for the Bernoulli.

To summarize, when drawing a topic assignment for a word, we either draw from the document tree (local DP) with probability $\frac{n_d}{\alpha+n_d}$ or we draw from the corpus tree (global DP) with probability $\frac{n_w}{\beta+n_w} * (\frac{\alpha}{\alpha+n_d})$ or else 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 until convergence.

For comparison, the equivalent step for LDA using our implementation of Gibbs sampling would require to sample only from:

$$z|w, d \sim \sum_k \frac{(A(k|d) + \epsilon) * (A(k|w) + \phi) * \delta_k}{(A(k) + (\phi * V)) * n_d}$$

3.4. How our model differ from nHDP

There are a few noticeable differences between HTMOT and nHDP in terms of the training process. First, we do not have a particular initialization protocol. The algorithm just starts with all words not assigned to any topic. Hence, initialization is similar to the rest of the training procedure except that unassignement is not needed. Second, we do not make use of a greedy algorithm to select sub-trees for each document. The tree for each document is created organically as the Gibbs sampler progresses. Our training algorithm is thus simpler and easier to implement.

Our implementation is available on github[6].

4. Experimental setup

4.1. Dataset

To perform our experiments, we crawled [7] 62k articles from the Digital Trends [8] archives from 2015 to 2020. This news website is mainly focused on technological news but also contain general news. It includes news about hardware, software and related companies but also space exploration and COVID-19. For all articles, we extracted the text, title, category and timestamp.

[6] https://github.com/JudicaelPoumay/HTMOT

[7] The crawling was performed using Python with the help of the BeautifulSoup library.

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

We discarded articles from the "deals" category as they cannot be considered as news articles, but are instead advertisements containing mostly URLs. We also discarded articles containing less than 500 characters. For the remaining articles, we concatenated the title and text.

Next, we removed common editor's sentences such as "*we strive to help our readers find the best deals on quality products and services, and we choose what we cover carefully and independently.*." and common journalistic phrases such as "*digital trends has reached out to*". Various other pre-processing steps were also applied:

- We extracted entities with the following tags using Spacy's NER:
  - Person, Norp, Fac, Org, Gpe, Loc, Product, Event, Work_Of_Art, Law, Language.
- Non-alphabetical characters were discarded.
- POS was applied on the rest of the document and only words with the following tags were extracted:
  - ADJ, NOUN, VERB,INTJ, ADV.
- Words with the "PROPN" POS tag were also extracted to cover some entities that spacy's NER missed.

Finally, infrequent words were removed if they occurred more in one document than in the rest of the corpus.

A good pre-processing procedure is essential for the interpretability of topics as shown in [32]. We believe interpretability is of primary importance for topic models which is why we extracted entities to help understand topics. 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. This greatly impacts the interpretability of topics as the entities shows actors in the topic such as personalities and companies.

We perform our experiments on two datasets, viz: the full dataset of 62K news articles and a subset containing articles only of the previous year (i.e. 2020) with 5k articles. Our aim in doing so was to compare the convergence speed with respect to the dataset size. Additionally, focusing on a specific year provides a more focused and fine-grained analysis for environment scanning. Hence unless otherwise stated, all results presented will be about the subset corresponding to the year 2020.

4.2. Parameters

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

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.

- Δ (depth time multipliers) : define for each depth a time coefficient which affects how localized topics are (see section 3.2).
  - For each depth we can set the coefficient to > 1, this disables time for that depth
  - Δ_4 defines the time coefficient for depth 4 and deeper depths.
- θ : how likely we are to create deeper sub topics.
- Θ : the importance of the Bernoulli prior.
  - Precisely, the priors for the Bernoulli distribution are $\theta_1 = \theta * \Theta$ and $\theta_2 = (1 - \theta) * \Theta$.

Second, we have parameters that regulate the growth of the trees.

- CM (Critical Mass) : the minimum valid topic size; only valid topics are outputted.
- SM (Splitting Mass) : the minimum topic size to 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.

Third, we have the Dirichlet prior parameters.

- ϕ : the prior for the topic-word distribution.
- ϵ : the prior for the corpus/document-topic distributions.
  - Precisely, these distributions refer to the base distributions used by the corpus/document-topic DPs.

Finally, we have training parameters.

- The batch size : the number of documents seen per iteration[9].
- Iterations[10] : 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 2 defines the value of each parameter for the models we will present in the next section.

[9] Its sole purpose is to help graph the various statistics by averaging over the batch.
[10] In the code Iterations=Iterations*epochs. Epoch only defines when to save checkpoints.

Name	Value	Category
α	0.00005	IDTs parameters
β	0.0002	
Depth Time Multiplier (Δ)	[2,2,0.2,0.2]	
θ	0.25	
Θ	1	
Critical Mass (CM)	0.0005	IDTs growth control
Splitting Mass (SM)	0.005	
Time To Live (TTL)	2	
ϕ	0.1	Traditional LDA topic parameters
ϵ	1	
Iterations	4500	Training parameters
Stop Growth Iteration (SGI)	2000	
Batch size	500	

Table 2: Parameters used for our model

5. Results and Discussion

We will now present our results. We will start by providing a statistical analysis of the extracted topics and of the training behaviors of the model trained. Then, we will discuss the results of the Word Intrusion task, its flaws and directions for future topic modelling evaluation methods. Finally, we will examine the various extracted topics qualitatively.

5.1. Interesting Observations

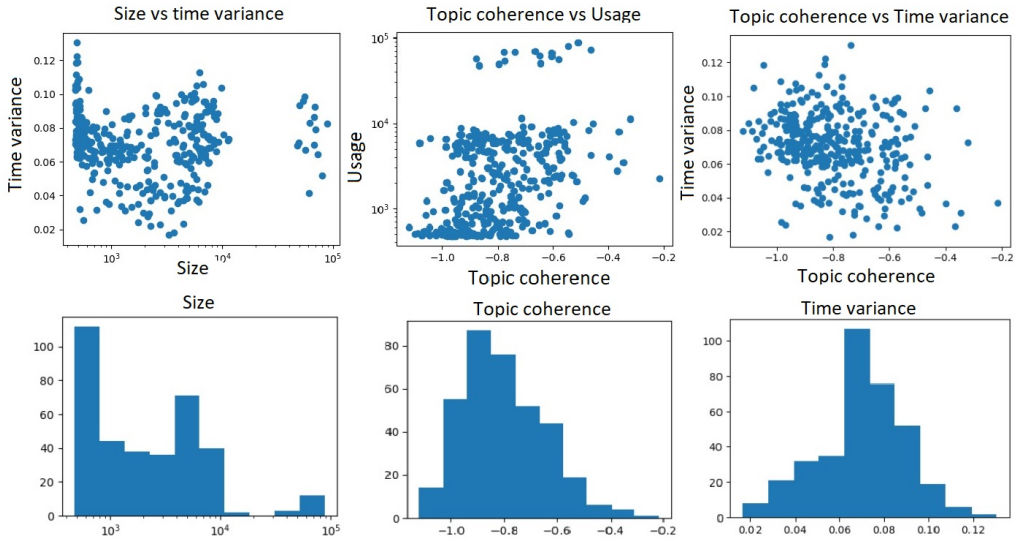


Figure 4: Exploratory analysis of the resulting topic time variance, size and coherence.

First, we will first take a look at topic statistics: time variance, size and coherence (see figure 4). While both time variance and coherence look normally distributed (although skewed), size is closer to an exponential distribution. For the topic size, this is the result of both the stick breaking construction of the Dirichlet Processes (DPs) and the fact there

Dyads	Pearson correlation coefficient
Size and coherence	0.25
Coherence and time variance	-025
Size and time variance	0.06

Table 3

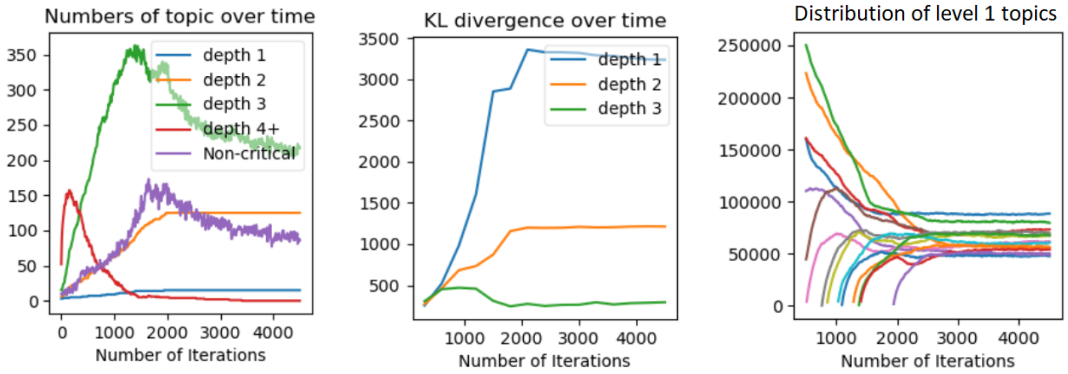
are many more sub-topics than the larger parent topics. As can be seen in table 3, topic size vs coherence and coherence vs time variance are noticeably correlated. Conversely, size and time variance are not correlated. Hence, topics that are larger or more localized in time tend to be more coherent. For the latter, this is because time localized topics tend to be about a narrower theme with a more specific vocabulary. A similar observation was done in Wang and McCallum [6].

However, for the former this is surprising as we expect smaller more specific sub-topics to be more coherent than larger parent topics which encompass a wider variety of domains. This paradox can be explained by the mathematics of coherence measures. Coherence measures ignore topic-word assignments and use co-occurence for all instances of the words. This implies that if a word is one of the top words (most probable words) of a small and a large topic, coherence measures will use all instances of the word to measure coherence and not just instances that were specifically assigned to that smaller topic. Thus, smaller topics might overshadow by larger ones as many instances used to measure coherence are more likely to belong to the larger topic than the small one. In other words, there will be more situations where that word will occur in the context of the larger topic than smaller one. Using the Umass formula of $\frac{p(w_1,w_2)}{p(w_1)*p(w_2)}$, this means the denominator will be disproportionately greater than the numerator. In a similar vein, if a small topic has a very specific vocabulary (rare words) but one of its top words is common this will disproportionately and negatively affect its coherence. For example, let say the topic of "engine testing" has the following top 5 words (test,fire, altitude, prototype, engine). While the words (altitude, prototype, engine) are quite specific, the words (test,fire) are much more common. Hence, considering the Umass formula and given that w_1 = "test" and w_2 = "altitude" then $p(w_1) >> p(w_2) > p(w_1, w_2)$ and consequently the coherence will drop dramatically. However, subjectively, there is nothing inherently incoherent with this pair of word for the "engine testing" topic.

The foregoing discussion highlights two limitations of the coherence measures. 1) By ignoring the specifics of topic-word assignments, larger topics overshadow smaller ones. 2) Topics with which have a mix of rare and common words will inevitably see their coherence score be penalized. These limitations are aggravated by hierarchical models like ours as we extract many small sub-topics which tend to have a mix of common words and more specific rarer words. Hence, coherence measures which are not based directly based on word co-occurrence might be preferable.

Another interesting observation pertains to the training behavior of the model is also interesting, see figure 5. The sub-figure (a) presents the number of topics over time. We can see that creation of topics at depth 3 and 4 peaks quickly and then falls; no depth 4 topics remained at the end of the training process. Conversely, the growth of depth 1 and 2 topic were stopped early, i.e. they plateau, based on the SGI parameter (see table 2). Sub-figure (b) shows the average KL

divergence between sibling topics at different depths. We can see that depth 1 topics are the most distinct. Conversely, depth 3 topics are closer in term of KL divergence; this make sense as they are conceptually related by a parent topic. Finally, sub-figure (c) shows the evolution in size of depth 1 topics. It can be observed that they converge quickly to a final size in a few thousand iterations but this convergence can be greatly perturbed by the creation of new topics which is another reason for the SGI parameter.



(a) (b) (c)

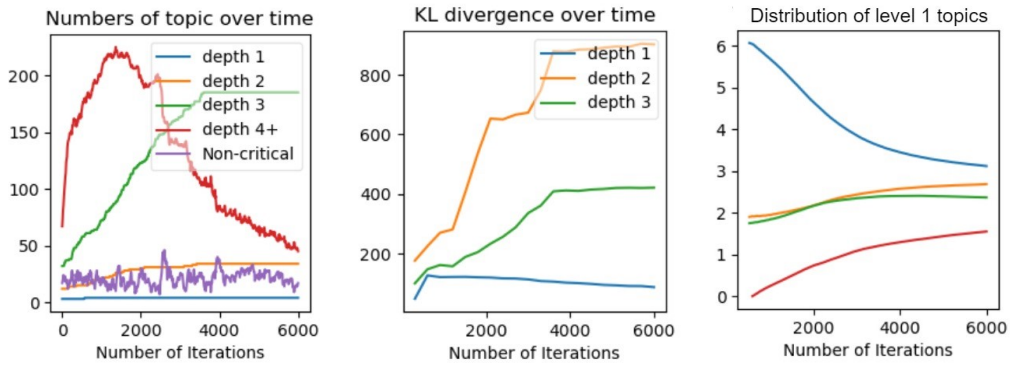
Figure 5: Training behavior for the 1 year model.

The aforementioned results describe the behavior of our model on the subset of 5k news articles for the year 2020. We repeated the same experiments on the entire corpus, spanning 2015-2020 (see figure 6). We observed that in this setting, fewer topics are created at the first depth. This can be explained by the difference in dataset size. As we can see in the equation 3, the probability of creating a new topic is $\frac{\beta}{\beta+n_w} * \frac{\alpha}{\alpha+n_d}$. Hence, as the dataset gets bigger, so does n_w (the number of instances of word w in the corpus) and the probability of creating a new topic gets smaller. As can be expected, the convergence speed is slower than when using the 1-year dataset as depicted by the sub-figure (c). However, the reduction in speed is not proportional to the difference in dataset size, illustrating the efficiency of our novel Gibbs sampling procedure. Specifically, the full 5 year dataset is made up of 62k documents, while the 1 year subset is made up of 5k documents. Thus, assuming a linear convergence rate, one would expect the number of iterations before convergence for the full dataset to be at least 10 times longer than on the 5k subset. However, as can be seen from the sub-figure (c), 3000 iterations were more than enough for the 1-year dataset and at 6000 iterations the 5-year dataset is already flattening. This indicates a sub-linear convergence rate which is highly desirable.

Considering the overall training time, our Gibbs sampler can go through 100k documents per hour on a single desktop computer[11]. This is comparable to the SVI algorithm used for nHDP (as described by [4]) in terms of speed : 90k articles per hour [12] [4]. Overall, our model (based on the 1 year dataset) was trained in $\sim$ 22h but converged after $\sim$ 14h at 3000

[11] Ryzen 5 3600x, 32Go RAM, NVMe SSD

[12] No information about hardware is provided, and language environment differs :MATLAB for SVI vs Python for our Gibbs sampler



(a) (b) (c)

Figure 6: Training behavior for the 5 year model.

iterations. The algorithmic complexity is linear with respect to the dataset size. However, the depth of the topic trees and thus the growth and regulating parameters for the Infinite Dirichlet trees can greatly impact performance. Nonetheless, as we have seen, convergence rate seems to be sub-linear. Hence, ten times the data does not imply ten times the number of required iterations.

5.2. The Word Intrusion task and its flaws

We applied the Word Intrusion task to evaluate our model. Originally, to asses the quality of a given topic, the Word Intrusion task involves selecting an intruder word from another topic. However, since our model is hierarchical, one could easily bias the Word Intrusion score by choosing words from topics that are most distinct, i.e. non-sibling topics. To avoid this caveat and to prevent the score to be biased in our favor, we only selected intruder words from sibling topics. The reasoning is that we need to evaluate if siblings, in particular, are distinct. For example, when selecting an intruder word for the sub-topic of ”astronomy”, we would choose from one of its siblings such as the ”astronaut” topic instead of a cousin topic such as the ”Covid-19 vaccines” topic which would have lead to a much more obvious intruder. (see figure 7).

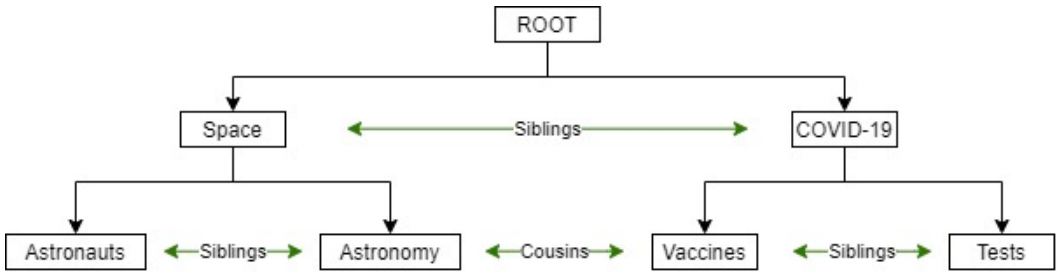


Figure 7: Example of a topic tree with cousins and siblings.

The survey used to perform this task was created using Google Forms [13]. The annotators come from a small internet

[13] It is also available on github

community that centers around sharing and answering surveys [14]. 57 respondents answered the survey over the month of may 2021.

Results show 74.83% accuracy in the Word Intrusion task (as defined in section 2.3) on 6 topics in various depths. This is on par with performance shown in [23]. However, deeper topics show worst performance: 98.25% for depth 1 topics and 63.13% for depth 2. We have also measured the confidence level of annotators: 92.63% confidence for depth 1 topics while depth 2 topics show 71.13% confidence.

Our results show that our model outperforms the only other hierarchical model evaluated with Word Intrusion presented in Pujara and Skomoroch [5] as it reports 27% Word Intrusion accuracy. This difference can be explained by the fact that their method requires the manual settings of the size and structure of the extracted topic tree. Specifically, they choose to set the number of topics and sub-topics to be 5. On the other hand, our model creates as many or few topics/sub-topics as it needs which makes it more flexible and adaptable. Moreover, [5] lacks a pre-processing procedure and elements such as numbers end up as topic words. This makes it more difficult for annotators to correctly find the intruder word. Thus, this observation illustrates the benefit of the pre-processing step of our model, which improves the interpretability of the generated topics.

Comparatively, the performance of non-hierchical models such as LDA, is around 85-90% accuracy; exact figures were not given [23]. For depth 1 topics, our model sees higher performance. Nonetheless, this can be explained by the difference in corpus used as LDA is similar enough to HTMOT for depth 1 topics. Nonetheless, as we have seen for HTMOT the accuracy of the Word Intrusion tasks quickly drops with topic depth.

However, the Word Intrusion task has some noticeable flaws, especially for hierarchical topic model. As we go deeper in the topic tree, sibling topics are increasingly similar. E.g. the sub-topics (astronaut, crew, launch, rocket, space) and (launch, rocket, space, satellite, payload) can be understood as being about astronauts and satellites in the wider topic of space. However, we can see that these topics share some important words. In other words, deeper sibling topics essentially share a smaller effective vocabulary. Figure 8 provides another example of this situation where the sub-topic intruder word, i.e. galaxy, is less obvious than that of its parent, i.e. title. Specifically, the deeper we go, the closer the intruder word will be to other words in a given topic. This is problematic because topics can be close (in term of KL divergence) but still distinct. Hence, the Word Intrusion tasks is less reliable for deeper closer topics. We argue this is because it is a topic centric evaluation method and it lacks a global view of the topic tree. Finally, hierarchical topic models can potentially produce hundreds of topics. Thus, creating a survey assessing a significant part of them is not feasible as the survey needs to be small enough for people to be willing to answer it.

The Word Intrusion task has other flaws. It is a binary classification which is a coarse-grained approach and does not indicate the level of confidence of annotators. It would be preferable to ask users to rank words from most likely to least

[14] https://www.reddit.com/r/SampleSize/

Topic 1	Subtopic 1-1
mission	satelite
launch	launch
title	mission
planet	galaxy
space	rockets

Figure 8: Example of topics with intruder. On the left we have the parent topic and one of its children on the right. The intruder is highlighted in orange. Galaxy is a less obvious intruder than title.

likely to be an intruder. However, this would be a heavier process. Furthermore, extracted topics can span a wide range of themes. Hence, annotators may not be knowledgeable about all of these topics; thus making it more difficult for them to find the intruder. When running our experiments, one annotator actually remarked on the difficulty of not recognizing some words which hindered their ability to answer the survey. Another study made a similar observation when applying LDA to medical text [33]. Furthermore, this situation is worsened for hierarchical and temporal topic modelling as deeper topics tend to be lexically closer which requires even more knowledge to distinguish.

In lieu of these issues, we argue that the search for a reliable evaluation method for topic modelling must continue. We posit that at least four measures of performance are needed to evaluate topics: *coherence, distinguishability, clarity*, and *stability*.

Coherence can be defined as a measure of how much a group of words make sense together and can be evaluated using topic coherence measures; with the aforementioned limitations. Distinguishability can be defined as the ability to distinguish one topic from another; it can be measured using probability distance measures such as KL divergence. However, as we have discussed earlier, deeper topics will be closer in term of KL divergence but can still be distinct. Hence, we need a way to normalize the KL divergence to compare topics across depths. The size of the effective vocabulary of sibling topics might provide a good normalizing factor but more experimentation is needed. Stability can be defined as the variability in topic words through multiple run of the same model on the same corpus; it can be measured by quantifying the word overlap between the same topic through different run of the same model on the same corpus [34]. Finally, clarity can be defined as the level of subjective understanding of the theme of a topic and could be measured as the level of annotator agreement in a topic identification task (e.g. inter-annotator scores, such as the Kappa statistics). However, this is difficult and still requires a wide range of knowledge from annotators. For now, we believe examining topics manually still remains the best way to understand the performance of a topic model. However, this solution is prone to cherry picking. This is why our data and results are available on github alongside our code so that readers may investigate the results more deeply.

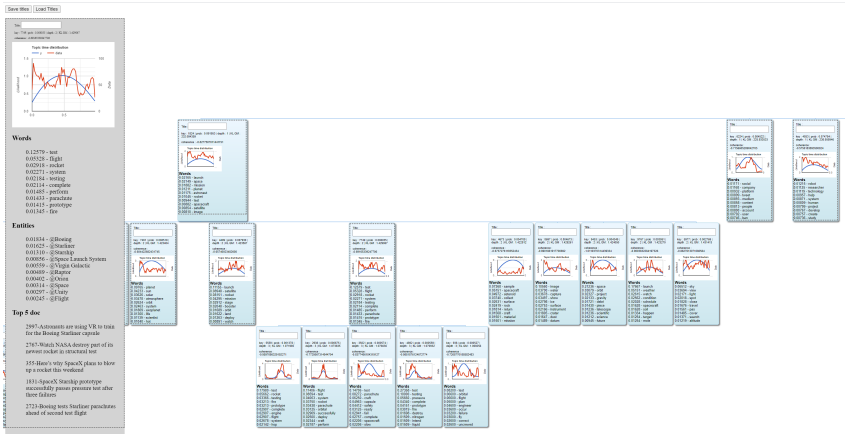


Figure 9: A snapshot of our output interface

5.3. Examining resulting topics

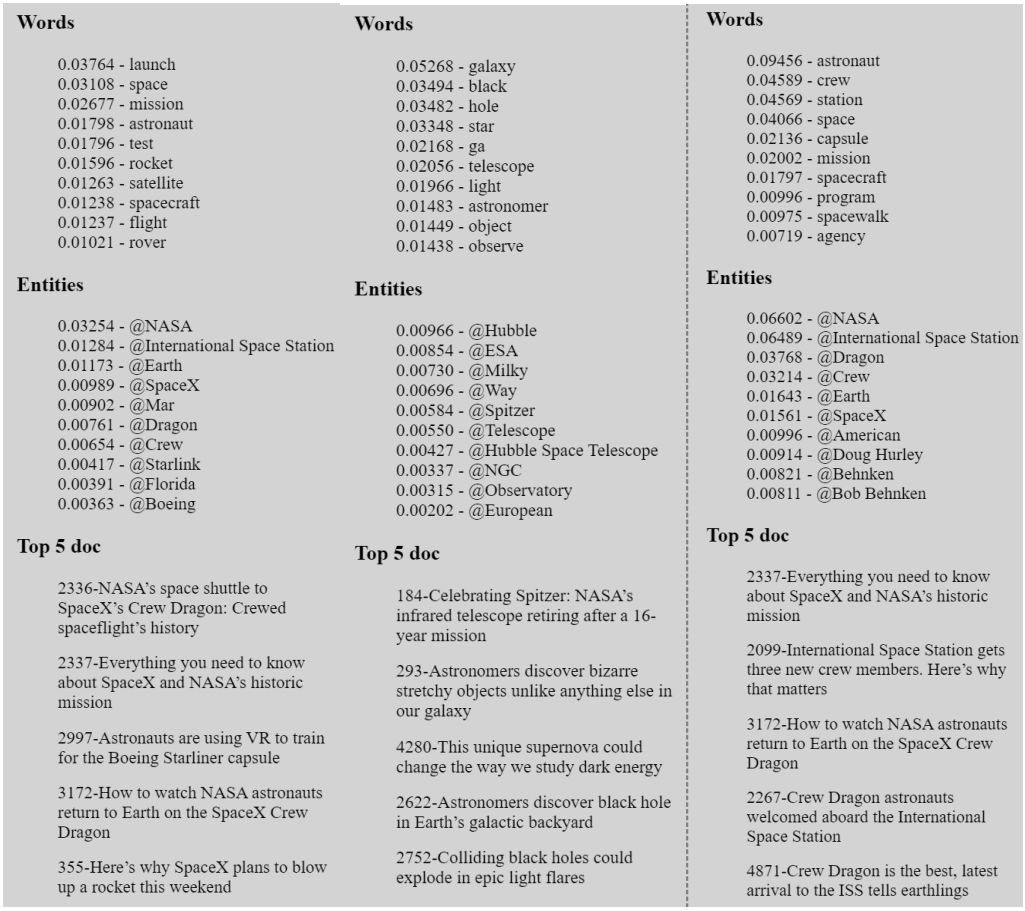
Since we modelled topic hierarchy (over time), we deemed it important to provide users with an interface for easy topic exploration and visualization, a snapshot of which is in figure 9[15]. Our interface provides:

- The ability to collapse and expand sub-topics.
- A representation of the estimated and empirical time distribution (blue and red curves respectively).
- The list of words, entities and the top 5 documents for each topic.
- A way to title topics and save them.

Now, we will inspect selected topics to illustrate the capabilities of our HTMOT model, i.e. whether it discovers meaningful topics. Specifically, we will focus on the topic of space exploration. Figure 10 presents a depth 1 topic (a) and two of its sub-topics (b,c). We can clearly see that the parent topic (a) is about space, and the two sub-topics are about astronomy (b) and astronauts (c), which are indeed related to the parent topic. This example also illustrates how entities can help interpret and understand these topics. For example, in the astronauts topic, we can see that Bob Behnken, (Doug) Hurley and SpaceX are important entities. A quick look at the top documents for that topic show that they were the first to fly on a SpaceX rocket. Moreover, in the astronomy topic, Hubble and Spitzer are frequent entities. This is coherent as they are two important low earth orbit telescopes. Other sub-topics of space include satellite launches, rovers, exoplanets, test flights, etc.

Figure 11, shows the topic of astronauts (itself a topic of the space exploration topic, as described above) and its three sub-topics: the return of Chris Cassidy (left), the launch of Bob Behnken and Doug Hurley (center) and the launch of three cosmonauts (right) as interpreted from the words, entities and documents associated with the topic. These topics

[15] A ready to use interface is available at https://github.com/JudicaelPoumay/HTMOT/tree/main/Results%20visualization%20example



(a) (b) (c)

Figure 10: The space topic and two of its sub-topics : astronauts and astronomy

are depth 3 topics and are well localized in time. The estimated time distribution of the sub-topics matches the timing of the aforementioned events: end of October 2020 (left), end of May 2020 (center) and the start of April 2020 (right).

Now, we will look at the document tree for one document, see figure 12. This was created by choosing only the topics that were assigned to at least 5% of words in the chosen document. The document in questions is titled "Astronauts are using VR to train for the Boeing Starliner capsule". The three main extracted topics are virtual reality applications, space and research. Two children of the topic of space were also assigned to this document: test flights and astronauts. Looking at the title alone, this tree encapsulate well the main themes of this document.

As we have discussed, we can use HTMOT to perform basic environment scanning in the space industry. We already mentioned the historic launch of Bob Behnken and Doug Hurley. Test flights are another subtopic of space (see figure 13). Its sub-topics are interesting as they show the different rockets that are being developed currently: the Boeing's Starliner, SpaceX's Starship, NASA's Space Launch System and the NASA's Orion spacecraft. This shows that the space race has been revived. Asteroid sampling is another sub-topic of space. Here we learn that such an endeavours are becoming more common with two spacecrafts from NASA (Osiris-rex and Lucy) and one from Japan JAXA (Hayabusa2) being mentioned in the top-5 documents and entities. These elements are only a sample of what we learned from the topics

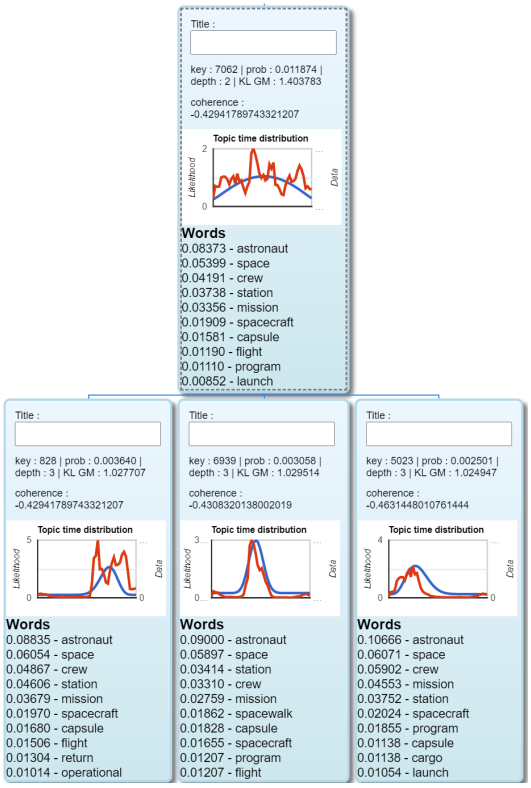


Figure 11: Examples of depth 3 topics localized in time for the 1y dataset

extracted by HTMOT. Thus, by itself our method already provides an efficient way to get insights about a specific domain or corpus.

5.4. Discussion

Furthermore, through a sensitivity analysis, we discovered that increasing the parameters α and β leads to wide and shallow trees. While smaller α and β lead to narrower and deeper trees. Overall, the same topics were extracted in both cases except that some topics that are depth 2 in the narrow/deep tree become depth 1 in the shallow/wide tree. Hence, a wider/shallower tree would separate astronomy and space exploration topics while a narrower/deeper would group those topics into a larger topic of space. However, it is not always obvious which configuration is better. If we look at the topic of space, we can see that it is mainly about space exploration. Astronomy, while it is related to space, is quite distinct compared to the other sub-topics extracted. Thus, the question arises, at which points two topics are related enough that they should be grouped and at which point are they distinct enough to be separated? We would argue that it is up to the user of the model to decide what kind of structures it would prefer. Narrow/deep trees have fewer topics per branch which might be preferable when trying to understand a corpus while wide/shallow trees require less computation.

Moreover, we experimented with the Δ parameter. We observed that the width of the time span of the estimated time distribution is proportional the value of Δ : dividing Δ by two decreases the time span of the topic by two (for $\Delta < 1$).

Finally, we observed that some depth 3 topics are not well localized in time. All of these have in common that their

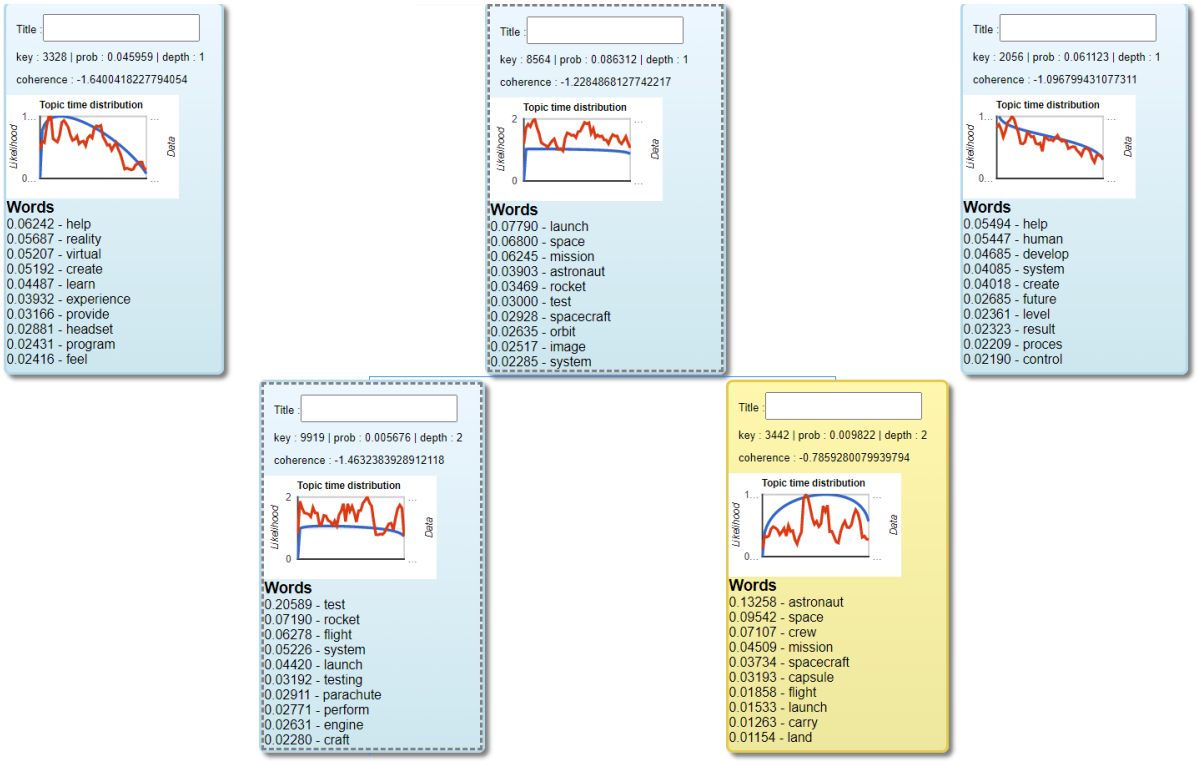


Figure 12: Example of document topic tree for the document : "Astronauts are using VR to train for the Boeing Starliner capsule" .

size is close to the CM parameter. We suspect that topics that are close to this boundary may fluctuate between valid and non-valid state. Since non-valid topics have their time modelling disabled, this might handicap such topic when modelling their time distribution. One solution would be to enable time when the topic size is $<$ CM. This would ensure that barely valid topics won't fluctuate between enabling and disabling time modelling. Nonetheless, it is also important to note that smaller more specific sub-topics do not have to be temporally specific which is another part of the explanation.

6. Conclusion and Future Work

We have proposed a new model for topic modelling capable of modelling hierarchy and time jointly as well as 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. Furthermore, we have discussed the flaws in the Word Intrusion task and coherence measures and pointed the need for a better and multi-factorial evaluation method, especially for hierarchical topic models. Moreover, we have shown how extracting entities can help interpret and understand topics at a deeper level. Finally, we have developed a tool to visualize topics, their hierarchy, temporality as well as their top words, documents and entities.

To experiment with our model, we performed an environment scanning of the space industry in 2020. Results show recent developments such as the new spacecrafts being developed (SpaceX's starship, Boeing's Starliner, NASA's SLS and NASA's Orion) or the growing number of asteroid sampling missions.

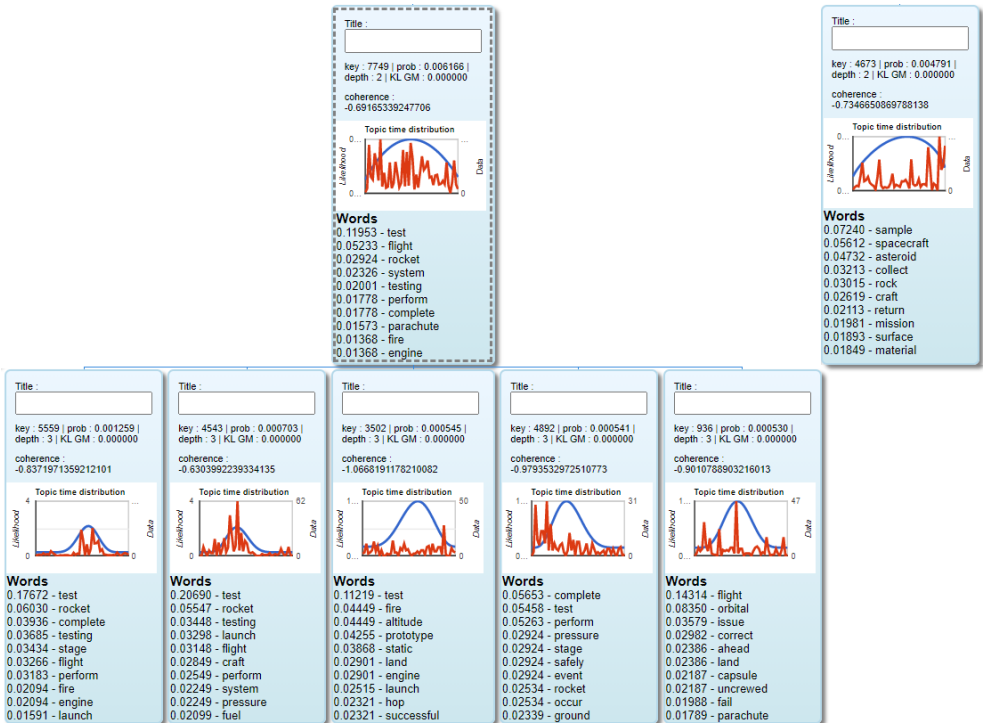


Figure 13: Test flight and its sub topics and the asteroid sampling topic

Future work includes developing better topic evaluation methods, experimenting with mini-batch stochastic Gibbs sampling to speed up the inference and incorporating topic correlation and/or sentiment analysis for a more complete topic model. Moreover, the beta distribution currently used to model temporality is mono-modal which does not reflect the complex reality of topic evolution. A mixture of beta or a more complex distribution could be used to improve upon the current solution by enabling time for high-level topics.

Acknowledgments

This research was sponsored by KPMG Belgium & Luxembourg through the HEC Digital Lab, HEC-Liège, ULiège. The funding source had no involvement at any stage of this research.

References

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[2] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical dirichlet processes, Journal of the American Statistical Association 101 (2006) 1566–1581. doi:10.1198/016214506000000302.

[3] D. M. Blei, T. L. Griffiths, M. I. Jordan, J. B. Tenenbaum, Hierarchical topic models and the nested chinese restaurant process, Advances in neural information processing systems 16 (2004) 17–24.

[4] J. Paisley, C. Wang, D. M. Blei, M. I. Jordan, Nested hierarchical dirichlet processes, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2015) 256–270. doi:10.1109/TPAMI.2014.2318728.

[5] J. Pujara, P. Skomoroch, Large-scale hierarchical topic models, in: NIPS Workshop on Big Learning, volume 128, 2012.

[6] X. Wang, A. McCallum, 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, Association for Computing Machinery, New York, NY, USA, 2006, p. 424–433. URL: https://doi.org/10.1145/1150402.1150450. doi:10.1145/1150402.1150450.

[7] D. M. Blei, J. D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd international conference on Machine learning, 2006, pp. 113–120.

[8] R. M. Nallapati, S. Ditmore, J. D. Lafferty, K. Ung, Multiscale topic tomography, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 520–529.

[9] Y. Song, L. Zhang, C. L. Giles, A non-parametric approach to pair-wise dynamic topic correlation detection, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 1031–1036.

[10] D. Mimno, W. Li, A. McCallum, Mixtures of hierarchical topics with pachinko allocation, in: Proceedings of the 24th international conference on Machine learning, 2007, pp. 633–640.

[11] Y. Wang, W. Xu, Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud, Decision Support Systems 105 (2018) 87–95. URL: https://www.sciencedirect.com/science/article/pii/S0167923617302130. doi:https://doi.org/10.1016/j.dss.2017.11.001.

[12] M. El Akrouchi, H. Benbrahim, I. Kassou, End-to-end lda-based automatic weak signal detection in web news, Knowledge-Based Systems 212 (2021) 106650. URL: https://www.sciencedirect.com/science/article/pii/S0950705120307796. doi:https://doi.org/10.1016/j.knosys.2020.106650.

[13] Y. Jung, Y. Suh, Mining the voice of employees: A text mining approach to identifying and analyzing job satisfaction factors from online employee reviews, Decision Support Systems 123 (2019) 113074. URL: https://www.sciencedirect.com/science/article/pii/S0167923619301034. doi:https://doi.org/10.1016/j.dss.2019.113074.

[14] N. F. Ibrahim, X. Wang, A text analytics approach for online retailing service improvement: Evidence from twitter, Decision Support Systems 121 (2019) 37–50. URL: https://www.sciencedirect.com/science/article/pii/S0167923619300405. doi:https://doi.org/10.1016/j.dss.2019.03.002.

[15] M. García Lozano, J. Schreiber, J. Brynielsson, Tracking geographical locations using a geo-aware topic model for analyzing social media data, Decision Support Systems 99 (2017) 18–29. URL: https://www.sciencedirect.com/science/article/pii/S0167923617300842. doi:https://doi.org/10.1016/j.dss.2017.05.006, location Analytics and Decision Support.

[16] H. Yuan, R. Y. Lau, W. Xu, The determinants of crowdfunding success: A semantic text analytics approach, Decision Support Systems 91 (2016) 67–76. URL: https://www.sciencedirect.com/science/article/pii/S0167923616301373. doi:https://doi.org/10.1016/j.dss.2016.08.001.

[17] H.-M. Lu, Detecting short-term cyclical topic dynamics in the user-generated content and news, Decision Support Systems 70 (2015) 1–14. URL: https://www.sciencedirect.com/science/article/pii/S0167923614002735. doi:https://doi.org/10.1016/j.dss.2014.11.006.

[18] E. B. Fox, E. B. Sudderth, M. I. Jordan, A. S. Willsky, A sticky HDP-HMM with application to speaker diarization, The Annals of Applied Statistics 5 (2011) 1020 – 1056. URL: https://doi.org/10.1214/10-AOAS395. doi:10.1214/10-AOAS395.

[19] X. Zhu, D. Klabjan, P. N. Bless, Unsupervised terminological ontology learning based on hierarchical topic modeling, in: 2017 IEEE International Conference on Information Reuse and Integration (IRI), 2017, pp. 32–41. doi:10.1109/IRI.2017.18.

[20] H.-C. Wang, T.-T. Hsu, Y. Sari, Personal research idea recommendation using research trends and a hierarchical topic model, Scientometrics 121 (2019) 1385–1406. URL: https://doi.org/10.1007/s11192-019-03258-x. doi:10.1007/s11192-019-03258-x.

[21] L. Hong, D. Yin, J. Guo, B. D. Davison, 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, Association for Computing Machinery, New York, NY, USA, 2011, p. 484–492. URL: https://doi.org/10.1145/2020408.2020485. doi:10.1145/2020408.2020485.

[22] X. Zhou, L. Chen, Event detection over twitter social media streams, The VLDB Journal 23 (2013) 381–400. URL: https://doi.org/10.1007/s00778-013-0320-3. doi:10.1007/s00778-013-0320-3.

[23] J. Chang, S. Gerrish, C. Wang, J. Boyd-graber, D. Blei, Reading tea leaves: How humans interpret topic models, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems, volume 22, Curran Associates, Inc., 2009, pp. 288–296. URL: https://proceedings.neurips.cc/paper/2009/file/f92586a25bb3145facd64ab20fd554ff-Paper.pdf.

[24] S. Ghosal, The dirichlet process, related priors and posterior asymptotics, Bayesian nonparametrics 28 (2010) 35.

[25] Y. W. Teh, Dirichlet process., 2010.

[26] R. Alghamdi, K. Alfalqi, A survey of topic modeling in text mining, International Journal of Advanced Computer Science and Applications 6 (2015). URL: https://doi.org/10.14569/ijacsa.2015.060121. doi:10.14569/ijacsa.2015.060121.

[27] B. V. Barde, A. M. Bainwad, An overview of topic modeling methods and tools, in: 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2017. URL: https://doi.org/10.1109/iccons.2017.8250563. doi:10.1109/iccons.2017.8250563.

[28] J. H. Lau, D. Newman, T. Baldwin, 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, Association for Computational Linguistics, Gothenburg, Sweden, 2014, pp. 530–539. URL: https://www.aclweb.org/anthology/E14-1056. doi:10.3115/v1/E14-1056.

[29] D. Newman, J. H. Lau, K. Grieser, T. 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, HLT '10, Association for Computational Linguistics, USA, 2010, p. 100–108.

[30] D. M. Blei, A. Kucukelbir, J. D. McAuliffe, Variational inference: A review for statisticians, Journal of the American statistical Association 112 (2017) 859–877.

[31] H. Xiao, T. Stibor, Efficient collapsed gibbs sampling for latent dirichlet allocation, in: M. Sugiyama, Q. Yang (Eds.), Proceedings of 2nd Asian Conference on Machine Learning, volume 13 of *Proceedings of Machine Learning Research*, JMLR Workshop and Conference Proceedings, Tokyo, Japan, 2010, pp. 63–78. URL: http://proceedings.mlr.press/v13/xiao10a.html.

[32] F. Martin, M. Johnson, More efficient topic modelling through a noun only approach, in: Proceedings of the Australasian Language Technology Association Workshop 2015, Parramatta, Australia, 2015, pp. 111–115. URL: https://www.aclweb.org/anthology/U15-1013.

[33] C. W. Arnold, A. Oh, S. Chen, W. Speier, Evaluating topic model interpretability from a primary care physician perspective, Computer methods and programs in biomedicine 124 (2016) 67–75.

[34] A. Agrawal, W. Fu, T. Menzies, What is wrong with topic modeling? and how to fix it using search-based software engineering, Information and Software Technology 98 (2018) 74–88.