ETH Search Project Report

Daniel Garellick
Department of Computer Science
ETH Zurich
Zurich, Switzerland
dgarellick@student.ethz.ch

Giulia Lanzillotta
Department of Computer Science
ETH Zurich
Zurich, Switzerland
glanzillo@student.ethz.ch

Andreas Opedal
Department of Computer Science
ETH Zurich
Zurich, Switzerland
aopedal@student.ethz.ch

1 Introduction

ETH is a world class research powerhouse, producing cutting edge insights on some of the most important challenges of our time, from Medicine and Data Science to Sustainability and Manufacturing. The current state of the ETH search bar however does not reflect to the fullest the state of the research at the institution, nor the successes of the people working in it. A query such as "Climate Change" to identify the leaders in the field for example, produces no results as depicted in Figure 1 below.

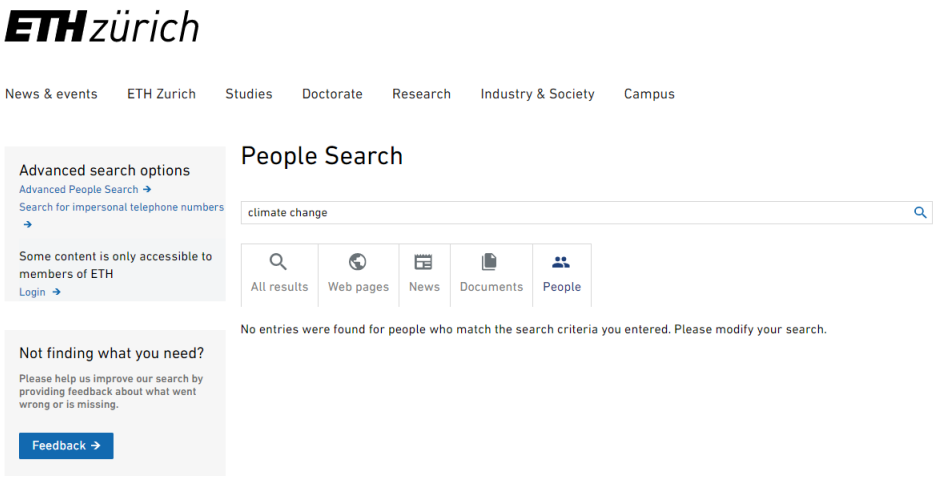


Figure 1: An illustration of the results from the query "climate change" in the ETH publications search bar.

Numerous projects have been launched to create a more reliable and more representative ETH Search, highlighting the achievements of the University and importantly providing insightful and relevant information to queries. The Data Science Lab project presented in this report is one of these projects, aiming to lay out the groundwork and hence creating the backbone for an improved ETH Search. By leveraging data from ETH publications, structuring it in the form of a Graph Database and gaining further insights by using Natural Language Processing (NLP) tools, we create a

solid proof of concept and propose a direction to follow for the future improvement of the ETH Search.

More specifically, the goals of the project are three-fold:

1. Unlock information present in the research collection

2. Identify and link publications, researchers and research areas

3. Generate valuable insight fir the ETH research community

In section 2 we describe the data that was used for the purpose of the project, the steps taken to pre-process it to extract the most valuable information and introduce the Graph Database to create the first links between the data. In the following section, section 3, we describe in detail the different models that were tested, how they relate to the requirements of our task, their shortcomings and their strengths. The final model designed to solution this task is novel to the best of our knowledge, combines insights from the various experiments performed and responds best to the requirements set out. Section 4 outlines the pipeline responsible for querying our model for relevant information and outputting a result. We conclude with possible improvements and extensions to the current model and a summary of the whole project in sections 5 and 6 accordingly.

2 Data Pipeline

One of the challenges of the project stems from the intrinsic variety of the provided data, both in terms of form and content. The ETH research collection contains multilingual tabular and text data spanning almost 100 years. Over the course of this time the data collection system underwent multiple substantial changes resulting in an overall incoherent structure. Further information on the ETH internal organisation has later on proven to be necessary in order to navigate the collection's tangled structure.

In order to facilitate the analysis, the collected data has been organised in the form of a graph. This particular choice will be detailed later in this section. In the next paragraphs, we describe the provided raw data, the data as it is represented in the graph database and the steps taken to get from the former to the latter.

2.1 Data Overview

A distinction that will accompany the following discussion is between *unstructured* and *structured* data, which in this case simply reduces to the publication abstract texts themselves and their corresponding metadata, respectively.

The abstracts were provided in a tabular format in the form of csv files, stratified by publication type. They came together with 113 other attributes relating to the publication, including e.g. author names, publication date and various identifiers[1]. Far from all of these attributes were relevant for the purpose of this project however, and a substantial filtering operation has been applied to the publication metadata, producing the data schema illustrated in Figure 2.

[1] A detailed version of the complete metadata schema is available here:
https://documentation.library.ethz.ch/display/RC/Metadatenschema

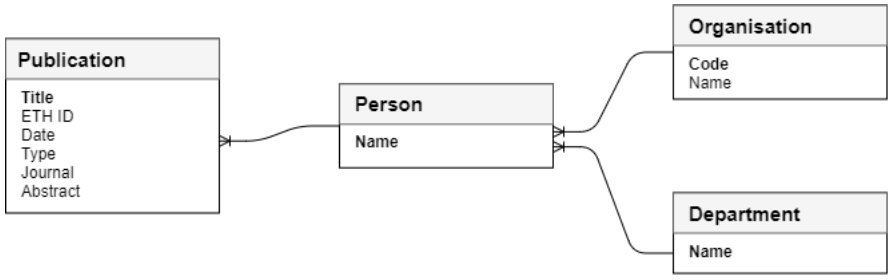


Figure 2: Operational data schema.

From the research collection metadata, information regarding not only each publication but their respective authors, and furthermore the organisations and departments they belong to was extracted. More specifically, as shown in Figure 2, we mined the authors' names, together with the name of their organisation and department (if present). Moreover, for each publication we obtained its *type* (e.g., *conference paper*, *book*, *journal article*, etc.) with the respective journal if present, its publication date and its title. Further metadata was obtained from two additional csv files: the first containing a list of all ETH professors with their organisation name and identifier, the second consisting of ETH department to *cost centre* mappings.

Additionally, a dump of the ETH professors' main website pages and the pages linked to them divided by department has been supplied. However, in the interest of time it was decided not to integrate the websites data, since the information to be gained was not deemed essential for providing a first demo of the the enriched search engine. Future work would involve leveraging the website data to validate the results from the modelling stage, as well as provide complementing information to the query engine.

2.2 Pre-processing

In order to prepare the data for import to the graph database and the modelling stage, the following pre-processing steps were required: attribute & subset selection, data cleaning and integration of the provided sources.

As anticipated above, the research collection exhibits characteristics stemming from multiple re-structurings accumulated over the years, namely: half empty fields, duplicated attributes, multiple non-unique identifiers. In addition, the data entry operation is not performed centrally and hence not homogeneously across the various departments and research groups at ETH, which contributes to the intrinsic variability in the attribute values: e.g., multiple formats for the author name or publication date. Consequently the leading principles for the pre-processing task were, not in any particular order, *completeness* and *extension*. The completeness of an entry can be identified as the number of relevant, non-missing fields, while the extension of a set of entries simply indicates its size.

First, we select a subset of the given publications by identifying the *publication type* values which present (a sufficient number of) abstracts, being journal papers, conference papers, book chapters and publications categorised as "papers". Filtering out all other publication types results in a dataset of $176,057$ publications. We note that only $21,421$ of these were complete with the abstract attribute. Entries not containing abstracts were also kept, reaching a compromise between the aforementioned extension and completeness. Additionally, for the text modelling stage, we chose to limit ourselves to abstracts written in English as the vast majority of the received abstracts were written in this language. Filtering out all other languages resulted in a corpus of $20,494$ documents.

Next, data cleaning steps were applied to the chosen subset in order to extract and unify the publication date and author fields to a single format. The author's name is used as key for the merge operation with the external metadata files integrating information on the university internal structure. It is therefore of critical importance for the success of the data integration procedure to carefully cast the variability in the name field to a unique, unambiguous string. In the interest of time our experiments in this regard have been limited to a few trivial regex formulas, which have nonetheless increased

the coverage (number of matching rows between the merged files) from $62,998$ to $68,999$. We are confident that by applying more sophisticated cleaning functions it is possible increase the coverage even further.

As anticipated above, the final pre-processing step consists in merging the data obtained from the research collection with organisational data from the two other sources. More specifically, we apply a *left* merge on the publication data, meaning that we keep all the entries in the selected subset of the research collection (even if no matching entry is found in the organisational data), but we avoid adding new ones. The final version of the research collection, to be exported into the database, comprises $877,216$ entries, with each entry corresponding to an author/publication pair, and the attributes shown in Figure 2.

2.3 Graph Database

The data gathered from the ETH research collection has been stored in a graph database. To our knowledge, a graph is able to best capture the structure in the organisational data. In addition, graph databases are optimised for *link traversals*, hence ensuring efficiency when querying for complex relational patterns. The importance of the latter feature will become clear in the section dedicated to querying. More specifically, the database was created in *Neo4j* (1), which has been chosen, among other reasons, for being a widely adopted technology and as such coming with the benefits of having an extensive documentation and a large community supporting it.

The output from the pre-processing stage above is translated into a first version of the graph containing nodes for Person, Department, Organisation and Publication, with relationships connecting the Person nodes to the others. Figure 3 gives an illustration of the graph in this stage.

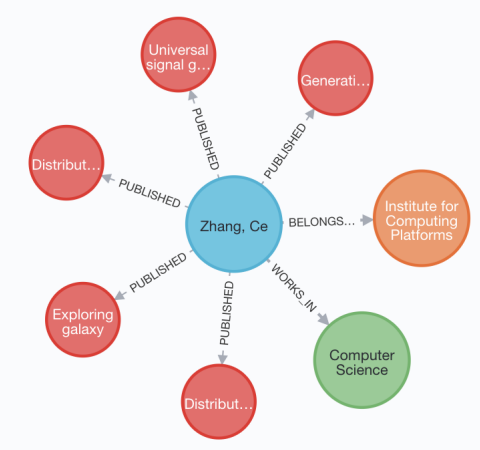


Figure 3: The results of an example query possible on a first version of the graph containing only metadata.

3 Modelling

The high-level goal of the modelling stage is to improve the ETH Search results, exploiting the information extracted in the pre-processing step. There are of course multiple ways to achieve such an open-ended goal. In the interest of time, our efforts have focused on a single line of development, consisting of enriching the graph with information gained from the unstructured data.

In a nutshell, graph enrichment consists of expanding the research collection graph described above with more refined features automatically extracted from the available data. Specifically, we rely on the unstructured information provided (i.e. the abstract texts) to identify the *topics* present in the collection. These are inserted as new nodes in the graph, with connections to the publications they describe and the most probable words they include. By integrating the raw data with more abstract concepts the idea is to create a "conceptual hierarchy" that can link the *meaning* of a query to the concrete information answering it.

We now proceed to describe the topic modelling stage, which plays a central role in the overall system and has therefore taken up a significant share of our efforts in this project.

3.1 Model requirements

To fulfil the goal of the project a list of prioritised requirements was created which suitable topic models would have to satisfy.

1. High Performance
2. Lifelong/Online Learning
3. Flexible Number of Topics
4. Automatic Topic Name Inference
5. Identifying Topic Hierarchies and Correlation

High Performance Performance is an essential requirement for a system that aims at bearing the backbone of the search engine for an international research hub such as ETH. Therefore *high performance*, unlike the next requirements, is a minimum qualification standard, that needs to be met by any candidate model. Unfortunately, unsupervised tasks such as the topic modelling implemented here, are difficult to tune and assess. There are no quantitative metrics corresponding directly to a better performing topic model and there are furthermore no standard set of qualitative criteria that represent a good topic model. With this caveat in mind, we regard a set of topics to be suitable if they lead to a reliable search and a graph that contains useful information. More specifically, the topics extracted needed to be representative of the different research areas embedded in the abstract data, while being mutually distinct and not too general in order to capture the wide variety of research published from ETH. A more detailed account of the techniques employed to evaluate the model will be provided in the next sections.

Continual Learning and Flexible Number of Topics Furthermore, as the research at ETH is continually evolving and papers, journals and books are published on a regular basis it is important for the model to be flexible to additions of new data and consequently topics. It is indeed unrealistic to expect a "fixed" model to accurately generalise to non-stationary content, as the publications for example reflect continually evolving research. In order to reflect this constant flow of new data, the model should permit periodic updates, integrating new unseen topics and modifying existing ones when necessary. It follows that such a flexible model must be free of the assumption common in topic modelling of having the number of topics fixed *a priori*.

Automatic Topic Name Inference and Topic Hierarchies/Correlation Additionally two, less pressing requirements have been identified, namely automatic topic name inference and finding hierarchies of topics or correlations between them. Often topic models describe the topics by the words they contain and thereby lack a systematic way to do topic name inference. Nonetheless, pairing each topic with a descriptive name could be a valuable asset in the prospect of gaining useful insights from the enriched graph. Secondly, topic models are typically non-hierarchical, meaning that they are not able to identify correlations and higher-level structures between topics. However, again with the purpose of increasing the interpretability of the end product, it would be beneficial to resort to hierarchy-oriented models. Furthermore, modelling topics in hierarchies could allow us to build a tree of research areas conducted at ETH at different levels of granularity. Coupled with automatic name inference, it would give valuable insights into the how research areas evolve and are connected throughout the institution. Finally, being able to model correlations in topics identified could assist performance in query time, as similar topics could be identified more efficiently and more accurately.

3.2 Pre-processing

There are a number of further pre-processing steps, not addressed in the data pipeline procedure, that might aid in building a high-performing topic model from the research abstract corpus. We first split the texts into lists of words by tokenising them. Here we choose the simple pre-processing function provided by the NLP open-source library Gensim, which transforms letters to lower case and removes accent marks. Following the tokenisation step, we delete stop-words, which are highly frequently occurring words that do not carry semantic meaning (e.g. "the", "at" and "which").

The remaining pre-processing steps taken depend on the model considered. For some models we might want to keep the words as they are, and for others we might want to e.g. group inflected versions of a word together so it can be considered as a single unit. A commonly used method in this regard is *stemming*, which simply "chops off" the ending of a word. As an example, the words "studying" and "studies" are mapped respectively to "study" and "studi". Stemming is computationally fast and easy to implement, but it does not capture irregular conjugations. For instance, "better" and "good" would not be grouped together after stemming has been applied. A different procedure is *lemmatisation*, which takes the context of a word into account and identifies its part-of-speech. Going back to the above example, lemmatisation would produce the same output "study", for both the terms. In our models we apply stemming and lemmatisation in varying combinations as part of the pre-processing, depending on the benefits entailed on the downstream task.

Another potentially helpful technique is to group words into bigrams, being two words that frequently occur together. For instance, the word "machine" and the word "learning" carry very different meaning if they occur independently or together as one unit. A coherent topic model, and furthermore a powerful search engine, will need to distinguish between these cases. The notion of bigrams can also be generalised to higher orders - in our experiments we include up to trigrams. The final model utilised neither bi- nor trigrams however, as will be explained below.

3.3 Experiments and Metrics

Topic modelling, originated as *latent semantic indexing* (Deerwester et al., 1990 (2)), is an unsupervised method for processing a set of documents, also referred to as *corpus* or *collection*, and identifying the topics that characterise it. The type of data analysed by topic models can come from a variety of sources and form, however the approach has witnessed most of its success on text mining and information retrieval, the former being the focus of this project. Topic models like the ones described in the following sections are generative models, in which documents are represented by a random mixture of latent topics and each topic by a distribution over all the words in the vocabulary.

In the following sections we present the various experiments conducted, their respective results and how these were compared leading up to the final topic model adopted for the ETH research collection. In this regard, we now briefly introduce the metrics adopted in the selection process.

Topic models are typically trained to maximise the dataset log-likelihood. However, as was touched upon above, a higher log-likelihood does not necessarily correspond to a better topic model as we qualitatively define it. In fact, likelihood-based evaluation metrics have been shown to not correlate well with human judgements of the quality of topic models, through tasks like word intrusion and topic intrusion (3). Furthermore, a log-likelihood score is somewhat meaningless on its own, which is why its use is relegated to model comparison. A popular alternative is *perplexity* , often preferred to log-likelihood for its interpretability:

$$PP(p) := 2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)}$$

where p is the density function of a discrete probability distribution and $H(\cdot)$ is the entropy. When applied to topic modelling it can be rewritten as the inverse likelihood of the test set, normalised by the total number of words. It is thus a decreasing function in the log-likelihood, meaning that a lower score corresponds to a better model.

In recent years a plethora of articles have been published investigating metrics better correlated with human judgment - (4), (5), (6). A widely adopted approach today relies on the concept of *coherence*. Topic coherence scores a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These measurements help distinguish between topics that are semantically interpretable and topics that are artifacts of statistical inference. This notion of topic coherence, being qualitative in nature, has led multiple different metrics to be developed. In this project the metric labeled C_V (introduced in (7)) has been adopted. It defines a sliding window used to retrieve co-occurence counts between words, which are then used to measure the normalised mutual pointwise information between the top words in every topic. This results in a set of vectors, one for each top word. The coherence metric C_V is taken as the mean over the cosine similarities between every top word vector and the sum of all other top word vectors. This metric is however considerably more computationally heavy than likelihood-based metrics and was for this reason used more as a final performance assessment tool than a proper selection instrument during the course of this project.

To reach a compromise between quality of topic assessment and computational feasibility, we take the following steps in order to select the best performing model:

1. Define a grid of hyperparameter values, one of which denotes the number of topics k.
2. Select the best model for each value of k by perplexity on a held-out test set.
3. Compare these models by "human judgement" and select the best final model according to the qualitative criteria discussed above.

3.4 Baseline Models

There are three baseline models identified in the attempt to address the aforementioned requirements to varying extent, namely *Latent Dirichlet Allocation* (LDA) (8), *Correlated Topic Model* (CTM) (9) and *Pachinko Allocation Model* (PAM) (10). The first one has been selected for being a well-known high performing model. First published in 2003, LDA has to date proven resilient to time as most new models are still compared in performance to it. CTM modifies LDA, identifying correlations between topics. PAM brings the complexity a step further by discovering a multi-level hierarchy of topics. None of the models considered here inherently fulfills our second requirement, however they have been chosen as baseline models for being relatively easy to train in comparison to more elaborate alternatives. The code to train and tune the model is built on the APIs offered by the library tomotopy, which has been chosen for its user-friendliness and completeness.

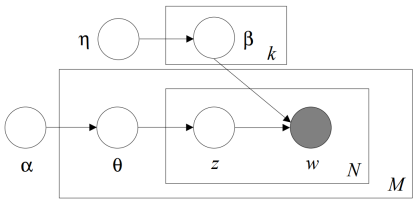


Figure 4: Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

LDA Latent Dirichlet Allocation, developed by Blei, Ng, and Jordan in 2003 (8), was the first generative topic model to be introduced in the literature. LDA regards documents as generated from randomised mixtures of hidden topics, which are seen as probability distributions over words. The generative process for each document can be summarised as follows:

1. Choose $N \sim \text{Poisson}(\xi)$
2. Choose $\theta \sim \text{Dir}(\alpha)$
3. Choose $\beta_z \sim \text{Dir}(\eta), \quad z \in \{1, ..., k\}$
4. For each of the N word positions n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$
   (b) Choose a word $w_n \sim \text{Multinomial}(\beta_{z_n})$

LDA models each document as a Multinomial distribution over the k available topics, and likewise each topic encodes a second Multinomial distribution (with one trial) over the vocabulary. Here, α and η are the hyperparameters of two Dirichlet distributions, specifying a prior over the document-to-topic and the topic-to-word distributions, respectively. The parameter β is a $k \times V$ matrix containing the probabilities for each word/topic pair, i.e. $\beta_{ij} = p(w_i = 1 | z_j = 1)$, whereas θ holds the topic probabilities for the document under consideration. It is important to note that k, the number defining the number of topics spanning the whole collection, has to be set in advance.

Figure 4 gives an illustration of the generative process with its associated parameters. Here, the rectangular 'plates' indicate replicate actions within the model, with the label of the plate corresponding to the number of replications. Circles represent variables or parameters, where white circles indicate

that the variable is latent or hidden and shaded circles indicate information which is given. The arrows demonstrate the hierarchy of influence of one variable over another. For example the plate N containing the variables z and w indicate that the topic will be individually and independently sampled from the distribution θ for each word of N words in the document.

The matrix β together with θ compose the learnable parameters of the LDA model. The learning phase requires computing the posterior of the model conditioned only on the hyperparameters α and η. However, this posterior is in general intractable to compute, leading us to resort to variational inference techniques, which only approximate the optimal solution and often exhibit an intrinsic stochasticity.

As suggested by the above, LDA does not meet the posed requirements other than (potentially) providing a high performance. Although it is a generative model, allowing it to infer topic distributions for unseen documents, it does not provide further updates to the parameters once already trained. Moreover it assumes the number of topics in a collection to be fixed a priori. It is essential in order to obtain a high performance from the model to carefully tune the parameter k to best match the collection as it is. Furthermore, as the hyperparameters α and η can significantly influence the features of the learned distribution, they are typically included in the model selection process as well.

The selection of values for α and η depend on the size of the vocabulary and the number of topics selected; Steyvers and Griffith (11) suggest $\alpha = 50/k$ and $\eta = 0.01$ as a broad choice which they have found to work well with a variety of text collections. In our experiments, we tuned LDA on a grid of different values for α, η and k. A new instance of the model is trained for each triplet of values on a subset of the available data, and its performance is then assessed on the remaining entries using the perplexity measure. A large range of values for k (up to 1000) were initially tested in a parameter search focusing only on k, but higher values beyond what was included in the grid below resulted in poor-quality topics. We choose k to take values in $\{50, 100, 150, 200, 300, 350, 450\}$, the entries of α to take values in $\{10/k, 1/k, 0.1/k\}$ and the entries of η to take values in $\{10/w, 1/w, 0.1/w\}$, where w is the number of unique words in the corpus. Note that in this way the two hyperparameters will take on values considerably lower than what advised in (11). Larger values of these parameters translates to a more uniform-like distribution over the topic weights θ (or equivalently the word weight β), indicating that documents consist of a larger variation of topics. In our corpus of research abstracts, we assume that each document is sampled mainly from a small subset of all available topics (accounting for the breadth of the research published from ETH). As such, we tune these parameters to take relatively small values. The scaling over the number of topics for α, and over the number of words for β is done in emulation of the above (11).

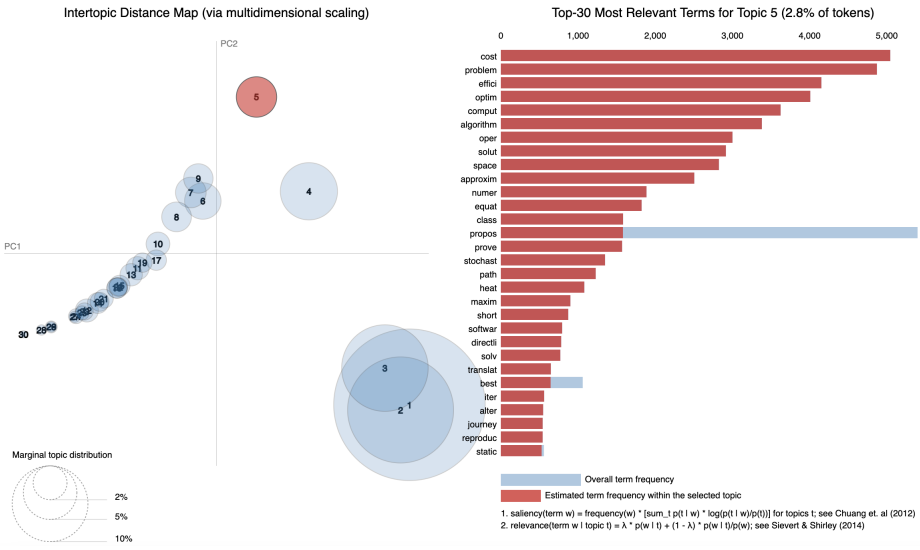


Figure 5: Visualisation snapshot of an LDA model containing 30 topics.

As mentioned in Section 3.3 the selection process is concluded by using human judgement to compare the best models according to the quantitative measures. To navigate the large body of information

available we leverage *pyLDAvis*[2], an interactive visualisation tool specifically developed for LDA models. Figure 5 displays a demonstrative screenshot of the tool for a model trained on 30 topics. The topics are projected to a two-dimensional space where overlappings of topics can be observed, shown on the left. Topics containing more general high frequency words are represented by larger circles and more specialised topics are represented by smaller circles. On the right side of the figure we see the most common words associated with a selected topic, which is compared to the overall frequency of the words (the blue bar, which is most often shadowed by the red bar since probable words in a topic are typically more likely to occur in that topic than overall). The particular topic shown contains (stemmed versions of) words like cost, optimisation, computation and algorithm, leading us to interpret this topic as theoretical computer science.

CTM The correlated topic model (Blei and Lafferty, 2005 (9)), shown in Figure 6, is very closely related to LDA. Its generative process is identical but for a change in the prior over the topic proportions θ. Namely, CTM replaces the Dirichlet distribution with a logistic normal distribution. This choice allows for dependencies in the form of correlations between the topics, determined by the covariance matrix in the logistic normal distribution. The failure of LDA to capture these dependencies stems from the independence assumptions implicit in the Dirichlet distribution on the topic proportions. Intuitively, this means that the CTM model can discover more realistic relationships between the different topics. For instance, the fact that one topic occurs frequently in a document might mean that other related topics are likely to occur as well. This is a trait that might very well be present in a corpus of research publication abstracts, as research fields overlap and co-occur in interdisciplinary publications. By modelling topic correlations, CTM has the potential for higher performance than what can be provided by LDA. In so doing it furthermore fulfills the last requirement specified.

CTM's additional expressivity comes at the cost of training time however. The most challenging element of this approach is computing the posterior, which is once again intractable. The problem is analogous to the one encountered with LDA, but fewer approximate inference algorithms are suitable for use in CTM as the logistic normal is not conjugate to the multinomial distribution. Like LDA, CTM assumes the number of topics k to be a known value and it is therefore crucial to conduct a grid search over the parameter space. We again also include the hyperparameter η (not to be confused with the vector of topic proportions η_d in Figure 6) of the Dirichlet prior over the topic-to-words distribution. We attempt to train the CTM model for k assuming values in the set $\{50, 100, 150, 200, 300, 350, 450\}$, and η in $\{10/w, 1/w, 0.1/w\}$, with w representing the number of words in the vocabulary[3]. However, due to a bug in the tomotopy implementation of CTM stemming from the sampling step of the inference procedure, we were not able to perform this grid search. Time constraints together with the fact that CTM did in fact not meet a majority of the requirements led us to not attempt an implementation on our own.

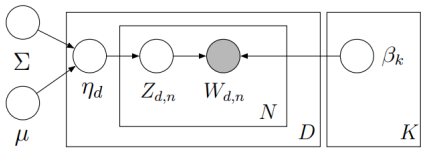


Figure 6: Graphical model representation of the correlated topic model (CTM).

PAM Finally, the Pachinko Allocation Model (Li and McCallum, 2006 (10)) shifts the classical topic modelling framework by creating a directed acyclic graph (DAG) mixture model in order to capture a variety of different kinds of topic relationships. In the basic structure of this model, the words in the vocabulary are represented by leaf nodes in the graph, with their parent nodes being distributions over these words. Note that also LDA can be represented by a two-level structured graph like this. Unlike LDA however, PAM allows for an arbitrary number of layers in the DAG. As such it not only models topics as mixtures over words, but higher level topics as mixtures over topics,

[2]Code available here: https://github.com/bmabey/pyLDAvis

[3]The larger span of values for the number of topics being grid-searched in this case is due to the lack of a preliminary optimisation over k as performed for LDA, which is in turn owed to the increased computational cost in training.

resulting in a hierarchy of topic relationships. Figure 7 gives a graphical illustration of this structure. Similar to LDA, Li and McCallum assign each interior node a Dirichlet distribution parameterised by a vector with dimensionality equivalent to the total children nodes which it encompasses. Documents are generated by first sampling from each Dirichlet to select a multinomial before generating each word within the document. The structure is highly flexible and depending on the layout of interior and leaf nodes can range in final form from a tree plot to an arbitrary DAG with complex features such as cross-connections and edges which pass over levels (as shown in Figure 7). It also demonstrates an ability to support significantly more topics than CTM and LDA, as shown by the authors of the original paper.

In short PAM has the potential to achieve a higher performance than the previously discussed models, while additionally satisfying at least partially the third set of requirements - modelling topic hierarchies. However, this approach suffers from two major issues. The first comes from the explosion in the dimension of the parameter space due to the fact that the multiple Dirichlet parameters α need to be optimised to match the data, resulting in more computation time than LDA. The second problem stems from the fact that the hierarchical structure is assumed to be known a priori. In practice this is hardly the case, creating a need not only for parameter tuning but also for architecture engineering. The space of the parameter search is therefore a lot larger than for the other two baseline models.

The PAM implementation offered by the tomotopy library only supports two level hierarchies. The authors do not have a publicly available implementation for the paper, and hence in the interest of time we have relied on the somewhat limited tomotopy version, deemed however sufficient for baseline performance assessment. In our experiments we therefore distinguish between the number of *super topics* k_1, being topics in the higher level of the hierarchy, and the number of *sub topics* k_2, being topics in the lower level of the hierarchy. We set k_1 to a fraction of k_2 based on the principle that a richer set of sub topics is generated by a larger set of super topics. Specifically we let k_2 vary in $\{100, 150, 200, 300, 350\}$ and for each such value we test different proportionality constants, namely $k_1 \in \{\frac{k_2}{5}, \frac{k_2}{10}, \frac{k_2}{20}\}$. Like before, we include the Dirichlet hyperparameters in the grid search. The parameters α and η parametrise the document-super topic and the sub topic-word distribution distributions respectively. Again we let the former vary proportionally to the number of topics, whereas the latter is defined as a fraction of the number of words in the dictionary. Specifically, α can take any value in $\{\frac{1}{k_1}, \frac{0.1}{k_1}, \frac{0.01}{k_1}\}$, while η varies in $\{\frac{10}{w}, \frac{1}{w}, \frac{0.1}{w}\}$.

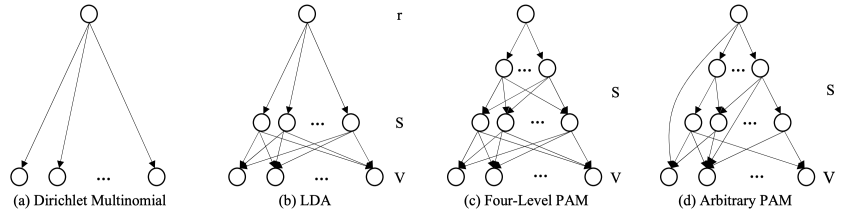


Figure 7: Structure of PAM model in various scenarios, showing LDA as a special case.

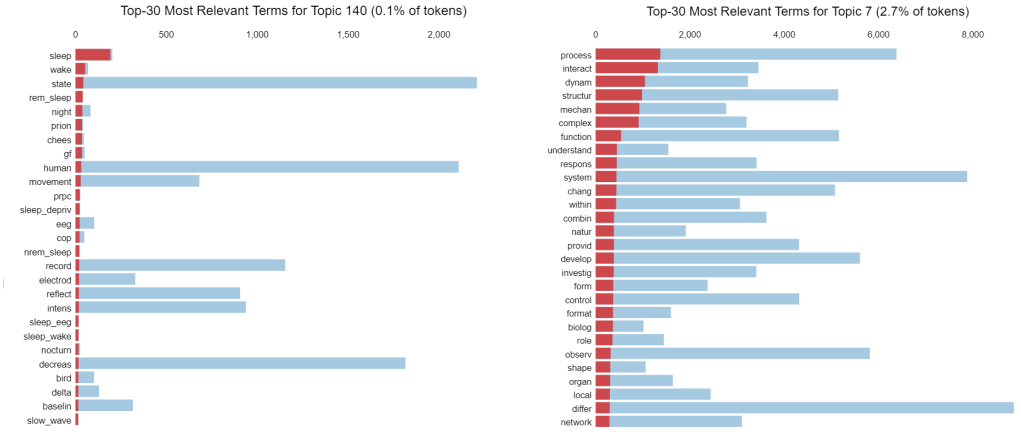
Results We now proceed to present the results of the baselines assessments as introduced above. In Table 1 the characteristics of the three best performing models are shown along with their perplexity score and average training time for both LDA and PAM, stratified by the number of topics (both super and subtopics in case of PAM). As can be observed, the perplexity score seems to be higher for larger number of topics for both models - the best performing model has 450 and $300 + 15$ topics for LDA and PAM respectively. We further note the considerable difference in runtime, where PAM experiences a training time in the scale of 10-100 times larger than LDA for roughly the same number of total topics. Together with the fact that it has a larger search space, this renders PAM very impractical to tune.

In addition to the quantitative overview of the results given above we include direct examples from the best performing model (LDA1) in order to provide the reader with a qualitative intuition of the baseline level. In Figure 8 we show once again a screenshot taken from the pyLDAVis interactive tool for two different topics extracted from the research collection. The chosen topics exemplify the observed characteristics in the model output. Both topics show high coherence, however topic 140 appears to be extremely specific, whereas topic 7 covers a more general concept. Likewise LDA discovers a relatively small set of general topics and a constellation of narrow but sensible topics.

Table 1: Baseline Grid Search results

Model	Description	PP	Training time
LDA 1	$k = 450, \alpha = 1/k, \eta = 10/w$	1.000023416	461 s
LDA 2	$k = 450, \alpha = 10/k, \eta = 10/w$	1.000023435	461 s
LDA 3	$k = 350, \alpha = 0.1/k, \eta = 10/w$	1.000023437	361 s
PAM 1	$k_2 = 300, k_1 = k_2/20, \alpha = 0.01/k_1, \eta = 0.1/w$	1.00156366	4,975 s
PAM 2	$k_2 = 350, k_1 = k_2/5, \alpha = 0.01/k_1, \eta = 0.1/w$	1.00156373	34,144 s
PAM 3	$k_2 = 300, k_1 = k_2/5, \alpha = 0.01/k_1, \eta = 0.1/w$	1.00156426	23,042 s

To our surprise the filtering step performed in the pre-processing has not been entirely successful in removing German text from the collection. However, with no additional aid, the model LDA1 has clustered together the remaining German words in a single "German" topic (see Figure 33 in the Appendix), effectively partitioning the two languages in separate topics. Further examples extracted from the same model are provided in the Appendix.



(a) LDA1 model Topic 140

(b) LDA1 model Topic 7

Figure 8: LDA1 direct examples.

3.5 Non-parametric Bayesian Models

In this section we look at the first set of experiments on topic modelling, which have been targeted towards the most pressing of our two secondary requirements, namely enabling continual learning and flexibility in the number of topics. In this regard, we have identified two non-parametric Bayesian models in the literature attempting to solve the above issues, namely *Hierarchical Dirichlet Process* (HDP) and its online inference version *Online-HDP* (O-HDP). Implementations for both the models have been made available by the authors, and further APIs for training and inference are offered by the well known NLP toolkit Gensim, which has been the choice for our experiments.

HDP The Hierarchical Dirichlet Process (Teh et al., 2006 (12)) is a non-parametric Bayesian model that can be used to model mixed-membership data with a potentially infinite number of components. Unlike its finite counterpart, Latent Dirichlet Allocation, the HDP topic model infers the number of topics from the data. It does so by sampling probability distributions from a Dirichlet process in a hierarchical fashion.

Simply put, in the generative process we distinguish the cases of sampling from an already existing topic from sampling from a new unseen topic - thus allowing a potentially unbounded number of topics. As topics are generated, there is a tendency to sample from topics which have occurred more frequently in the past with a higher probability. This clustering property can be understood better with the Chinese Restaurant Franchise analogy. Consider a Chinese restaurant with an unbounded number of tables, and n_j customers arriving for seating. The first customer gets seated at the first

table. All subsequent customers then get seated at an occupied table with a probability proportional to the number of customers already sitting there, and at a new table with a probability proportionate to some parameter α_0. When all customers have been seated we have a partitioning of the customers across all tables. This process can be repeated for J restaurants, forming a franchise. The analogy to topic modelling is as follows: Each of the J restaurants corresponds to a document, with the n_j customers corresponding to its words. Across all restaurants, the same menu of dishes, corresponding to topics, are served. The same dish is served to all the customers in a table, and each table is served only one dish. Thereby by analogy each word in a given document is associated uniquely with a single topic, and the customers sharing the same table represent words in the documents belonging to the same topic. The generative process of HDP is illustrated with example in Figure 9.

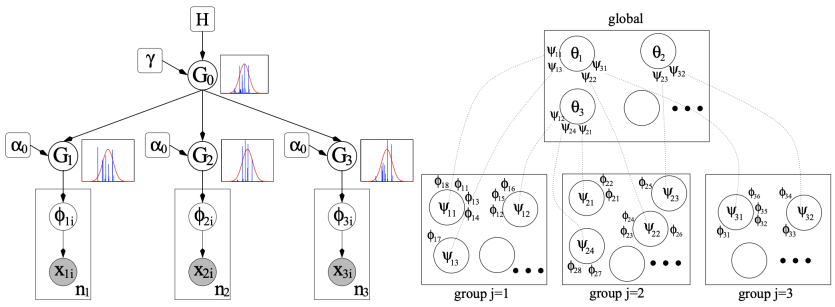


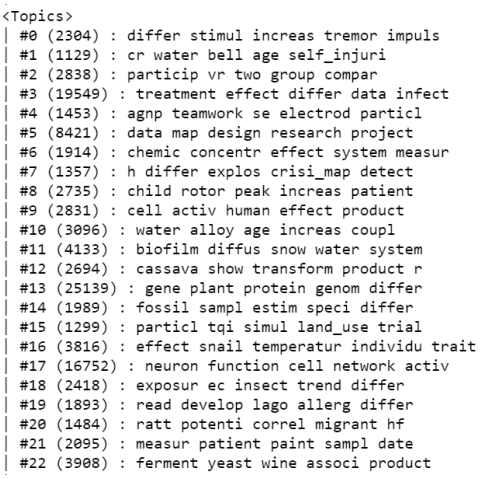
Figure 9: Illustration of the Hierarchical Dirichlet Process, with an instantiation of the Chinese Restaurant Franchise with three groups to the right.

Although HDP enjoys a significantly higher level of flexibility than the baseline models for what concerns its structure, it is still dependent on a number of parameters which are assumed to be known. Hence, albeit free from constraints on the exact number of topics, HDP still requires a prior on the ease of creation of a new topic. The performance of HDP proved to be intricately related to the tuning of this and other parameters as well as the pre-processing, which have been subjected to fine-tuning. To our dismay HDP did not outperform our baselines even after grid-search. Like before, in Figure 10 we show a random subset of topics inferred by the best performing model in order to provide an insight into HDP's performance. From a visual inspection of the results the extracted topics appear weakly coherent and noisy, often cluttered with general words and meaningless expressions. The drop in performance compared to the baseline level is evident. Even so, we do not draw the conclusion that HDP as a model is suboptimal in comparison to LDA. One factor that has a major impact on the experimental outcome is the intrinsic variability in the results, due to the approximate inference techniques adopted in the optimization of both methods. As such, the inference algorithm plays a significant role in the performance of the model. Thus we hypothesise that the use of different libraries for HDP and the baseline models can partially account for the observed gap in topic quality. Further support for this argument can be found in the tomotopy[4], where the authors offer a complete set of performance and runtime comparisons between their LDA implementation and the one offered by the Gensim package, with the Gensim implementation being outperformed in both aspects.

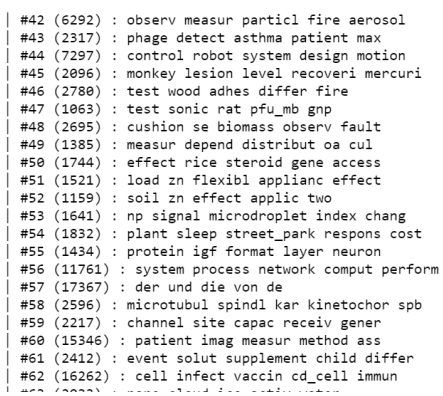
Nonetheless, the HDP model has demonstrated its usefulness by providing a point estimate for the optimal number of topics to be extracted from the collection. Specifically, the API requires a cap on the number of topics to be specified. The topics "in excess" are given probability 0 to appear in any document in the collection during inference. Setting such limit to 150 we observe 17 topics "in excess", thus the optimal number of topics for the research collection as discovered by HDP is 133. This estimate is to be taken with a grain of salt by the reader however, as is has shown to be remarkably susceptible to the hyperparameters values.

O-HDP Online HDP is the extension of the Hierarchical Dirichlet Process to online variational inference, for which an algorithm has been developed by Wang et al. and presented in their 2011 paper (13). The difference between HDP and its online version resides in the method adopted for approximate inference. The algorithm suggested by Wang et al. not only allows HDP to analyse much larger datasets, but it enables it to analyse streams of data. This last feature is the reason why

[4]https://bab2min.github.io/tomotopy/v0.10.0/en/



(a) HDP model topics subset 1



(b) HDP model topics subset 2

Figure 10: HDP model random subset of Topics. The topics are represented by the 5 words with highest probability under the topic distribution.

the O-HDP model has been considered among the candidate models for the research collection topic modelling task. Going back to our requirements, O-HDP potentially meets two out of three set of conditions by satisfying the need for both flexibility in the number of topics and continual learning.

Again we resort to the implementation offered by the Gensim library in our experiments. As for what concerns the hyperparameters, the values discovered by the grid search on HDP are chosen again, since this allows us to directly assess the effects of the new inference method. The online HDP model provides an inference and an update step. In order to assess the performance of the model on data streams we split our collection in two sets with varying proportions: $0.7 + 0.3$, $0.5 + 0.5$, $0.3 + 0.7$. We then proceed to perform an inference and an update step in this order, using the first fraction of the documents for the inference step, and the remaining ones for the update step. We compare the topics discovered in the first step with the topics identified by the HDP model, in order to evaluate the impact on topic quality of the inference algorithm and the decrease in number of documents. We then compare the topics inferred on the first set of documents with the ones revealed having integrated the remaining documents in the update. In order to better simulate the flow of data in a real scenario it has been necessary to perform a small change in the code provided. The implementation offered by the Gensim library relies on fixed data structures whose dimensions depend on the size of the vocabulary. However, we believe the assumption of a static vocabulary to be unrealistic for a research collection, where new concepts, and thus new unseen terminology, arise potentially within every publication cycle. For this reason, the O-HDP implementation has been extended to allow the vocabulary to grow together with the collection.

In Figure 11 two sets of topics obtained from the inference step performed on 70% of the collection are displayed. Two comments on these results ought to be made. Firstly, notice that the topics in 11a (specifically from topic 0 to topic 6) are hardly distinguishable in terms of meaning, as they all share the majority of their characterizing words. This phenomenon has not occurred in the HDP experiments, thus we believe it to be an undesired effect of the new inference algorithm. In order to assess the influence of this highly non-specific and homogeneous set of topics on the collection we look at the topic weights for each topic-document pair. The corresponding heatmap can be visualised in Figure 34 in the Appendix. The results are discouraging. In fact, most of the documents in the collection mainly involve almost exclusively the aforementioned set of generic topics (0 to 8 in the plot). For what concerns the rest of the topics (of which a random sample is shown in 11b) the results have all the characteristics observed in the topics inferred by HDP, namely low coherence, noisy entries and the presence of seemingly meaningless expressions. In fact, the performance of the online version of HDP appears to be strictly lower than of its precursor's. However, we shall keep in mind that the results shown in 11b are obtained by processing only a fraction of the entire collection, and hence a lower topic quality is to be expected.

```
Topic 0 -------
use - model - result - measur - system - base - data - studi - develop - differ - effect - method - user - test - design - incr
eas - process - ass - approach - show

Topic 1 -------
use - model - studi - data - measur - result - base - differ - effect - show - system - increas - process - develop - st - obse
rv - gener - also - compar - perform

Topic 2 -------
use - model - result - cpc - studi - concentr - chang - differ - activ - measur - base - effect - system - show - data - proces
s - ec - howev - present - method

Topic 3 -------
use - model - data - studi - differ - result - measur - system - base - show - effect - gener - observ - chang - present - anal
ysi - develop - method - compar - activ

Topic 4 -------
model - use - studi - differ - system - base - der - gener - effect - result - data - water - die - predict - measur - show - u
nd - develop - compar - perform

Topic 5 -------
use - model - differ - effect - result - studi - base - show - data - develop - system - perform - cell - method - activ - howe
v - compar - process - two - observ
```

(a)

```
Topic 38 -------
urban - analys - park - indic - studi - photograph - result - map - use - cdad - differ - measur - time - two - locat - pmmo -
definit - charg - pion - cancer - tumour

Topic 39 -------
product - qubit - effect - result - use - logic - input - voltag - growth - process - countri - show - sophist - ass - anomal -
isotop - taxat - overgraz - produc - orthorhomb - map

Topic 40 -------
base - high - iron - decomposit - neuropath - pain - algorithm - nerv - injuri - distribut - hemogen - object - three - transmi
ss - optim - iq - ferment - mouse - studi - age - mouse - progranulin - problem - price

Topic 41 -------
use - differ - bacteri - indic - anoli - tax - condit - gene - aa - vaccin - studi - follow - diagenesi - amino - antidepress -
micrantha - two - import - associ - composit

Topic 42 -------
region - flux - nearshor - km - model - calc - driven - co - product - dic - water - wheat - chang - biolog - air - sea - co -
simul - pco - variabl - yr - natur
```

(b)

Figure 11: O-HDP model pseudo-random subset of topics inferred from 70% of the collection. The topics are represented by the 20 words with highest probability under the topic distribution.

In Figure 12 we show the effect of the update step on the set of topics in Figure 11. We notice a group of generic and hardly distinguishable topics (displayed in 12a) almost identical to their pre-update version, whereas a substantial change at the semantic level can be observed in the remaining topics (see 12b). Further analysis confirms that the majority of the topics has likewise undergone radical changes with the update. As a consequence, all the document-topics distributions are completely altered as well. The implications of this behaviour are deleterious to our purposes. In fact, the model preceding an update and its successor would detect a different set of prominent topics for the old documents in the collection, which would in turn require adjustments on most of the database whenever a new set of publications has to enter the system (which is a common scenario in this application).

This concludes the experiments on HDP and its online variant, which, although potentially capable of doing so, have not satisfied our expectations and requirements. We turn next to embedding-based topic modelling approaches, which are the focus of the next section.

3.6 Embedding-based approach

Embedding based approaches have revolutionised many facades of Natural Language Processing (NLP) tasks, one of which is topic modelling. On a high level, embedding models work by encoding representations of words into high dimensional vectors, of size preferably smaller than the vocabulary size. The embedding is such that words with similar meaning are close to each other in this vector space.

Embedding-based topic models take embeddings one step further by integrating the topics in the vocabulary vector space. Consequently, these models naturally fulfil the topic name inference requirement. Being associated to a vector in an embedding space each topic can for instance be automatically labelled with the word closest to it, exploiting the intrinsic notion of distance provided by any vector space. But the potential benefits of embedding-based topic models are not limited to topic name inference. Pairing each topic with a vector representation leads to the potential of building topic hierarchies, e.g. by combining topics which are close to each other in the embedding space according to some similarity measure. The idea is illustrated in Figure 13, which depicts a cluster of topics related to sports as discovered by our chosen embedding-based topic model - the Embedded Topic Model (ETM) (14). Finally by leveraging word embeddings we can attend to the second of our requirements, namely online learning. More specifically, the idea is to fix a word embedding space and enrich this space with topics as more data (i.e. research publications) becomes available.

```
Topic 0 -------
use - model - result - measur - studi - data - differ - effect - system - base

Topic 1 -------
use - model - data - system - base - result - studi - differ - measur - effect

Topic 2 -------
use - model - base - data - measur - result - studi - differ - system - effect

Topic 3 -------
use - model - result - differ - studi - measur - base - data - show - function

Topic 4 -------
model - use - measur - studi - result - effect - differ - observ - base - process

Topic 5 -------
use - model - studi - result - data - base - measur - differ - effect - system
```

(a)

```
Topic 38 -------
np - energi - system - chang - ion - cost - product - process - household - use

Topic 39 -------
hon - use - approxim - observ - format - heartwood - model - glacier - charg - particl - possibl

Topic 40 -------
use - protein - express - scatter - strain - glacier - shop - trip - control - ascent - region

Topic 41 -------
drought - dea - ipsc - dispatch - use - dhi - reperfus - upon - indic - scenario

Topic 42 -------
catchment - dynam - increas - nitrat - streamwat - electr - miner - minimum - element - concentr
```

(b)

Figure 12: O-HDP same set of topics as shown in Figure 11 after the update of the model on the remaining 30% of the collection. The topics are represented by the 10 words with highest probability under the topic distribution.

Essentially, a new model is trained on each dataset, but the topics are added to the same space - resulting in a flexible system where new topics can be included as research fields evolve.

ETM The Embedded Topic Model extends the functionality of LDA by using embedding spaces (14). Indeed LDA proved to outperform other models described above and hence combining its advantages with embedding spaces could overcome some of the obstacles that LDA typically presents, while retaining its high performance. One scenario where LDA tends to perform poorly is when the vocabulary is large and "heavy-tailed", i.e. has a relatively large proportion of its probability mass concentrated by the tails which induces a more "uniform-like" distribution. We note that with a corpus of research publication abstracts stemming from a vast variety of academic disciplines, it is very likely that our vocabulary possesses these characteristics. ETM's ability to accommodate for larger and more heavy-tailed vocabularies further motivates experimenting with this model. An additional benefit of ETM is that it is more robust to stop words, eliminating the need to remove these during pre-processing. It does so by assigning topics in the area of the embedding space where such words occur and the stop words will therefore only be prevalent in the topics in their proximity.

The generative process of ETM is similar to the one of LDA. Assuming a static embedding of word representations encoded in an $L \times V$ matrix ρ and k topics represented by vectors α, it proceeds as follows to generate a document:

1. Choose topic proportions $\theta_d \sim \mathcal{LN}(0, \mathrm{I})$
2. For each of the N word positions n:
   (a) Choose a topic assignment $z_{dn} \sim$ Categorical(θ_d)
   (b) Choose a word $w_{dn} \sim$ softmax$(\rho^T \alpha_{z_{dn}})$

The key difference from traditional topic models is indicated by step 2b. Topics themselves are not encoded by distributions, but by vectors. Distributions over the words are taken as log-linear models, namely as a softmax over the inner product between the word representations and the topic. This places a higher probability of words in the proximity of the topic vector, but has, as topic mixtures in traditional models, support over all of the vocabulary. ETM also removes the Dirichlet prior

distribution from LDA in favor of the logistic normal distribution, as in CTM. This is done mainly for ease of re-parameterization in the inference algorithm, following the method presented in (15).

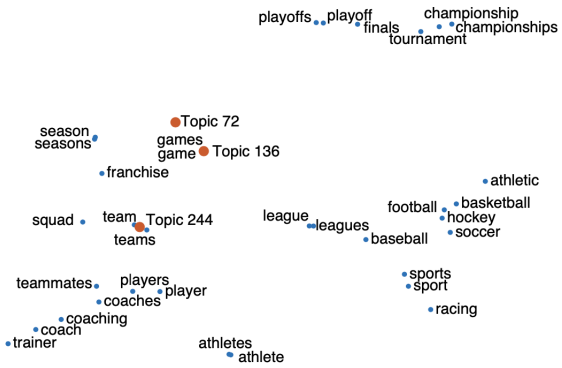


Figure 13: A cluster of topics related to sports discovered by ETM (14).

Word embeddings The ρ matrix encoding the word embeddings in ETM can either be learned during inference of the model or be taken as a pre-trained word embedding. Here, we choose the latter as we want a global embedding space to be used across batches. Furthermore, using a pre-trained embedding allows also for words not encountered in the training collection to be accommodated by the topic distributions, as the inner product between these words and the topic vectors can still be calculated. Multiple such pretrained embeddings exist and the two that were used for the purpose of this work were DistilBERT (16) and GloVe (17). The first is a smaller and hence faster version of BERT (18), a highly contextualised embedding. This implies that the vector of the word will be very much dependent on the context within which it was used. GloVe on the other hand is a static embedding that builds word vectors based on the word co-occurrence statistics of the corpus. Both were used to embed the corpus resulting from the abstracts.

Using DistilBERT in conjunction with ETM required significant pre-processing since ETM expects a fixed embedding matrix, mapping a term to a unique embedding vector, whereas any contextualised embedding produces multiple vectors for each word based on its context. We created a dictionary of words and their respective vector representations, where each word could occur multiple times according to the number of different semantic meanings it can have.

DistilBERT takes a word with its surrounding context and encodes it into a 768-dimensional vectors, reflecting the meaning of the word as it occurs in that context. See 14 for an illustration of words projected to a 3-dimensional space for visualisation. For example, the word mouse has different meaning in the sentences 'The cat chased the mouse' and 'I use my computer mouse as a pointer' and hence DistilBERT would create separate vectors for each. It does also, however, create an additional vector for 'mouse' in sentence 'The mouse ate the cheese', which should be close in proximity to the vector created for 'The cat chased the mouse'. In our dictionary we would ideally want one static embedding to encode both instances where mouse refers to an animal, and a separate one encoding the meaning of computer mouse. To achieve this, we analyse the distance between the created vectors for each word and if deemed similar, the vectors are collapsed into one[5]. We experiment with multiple methods such as cosine similarity and PCA to reduce the variability of the embedding. The cosine similarity method computes this metric between all pairs of same-word vectors, and collapses them to one if the similarity is above a certain threshold. The PCA method involves projecting the representations to the first principal component. This renders them static (19) but contrary to the aforementioned static embedding, the vectors obtained bear the summary of all the word's potential meanings. The two approaches (i.e., cosine similarity and PCA reduction) significantly differ in terms of memory requirements. Applying the cosine similarity filtering we keep from 1 to R vectors for each word in the vocabulary (where R is the number of occurrences of the word), hence producing a matrix of dimension at most size $V \cdot R \times E$, where $E = 768$ is the embedding dimension. For PCA we

[5]Following the ideas from the Standord AI Lab blog (http://ai.stanford.edu/blog/contextual/) and a tutorial by Chris McCormick (https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/)

force each word into a single vector, drastically reducing the embedding matrix dimensions to $V \times E$. Furthermore, we have devised the algorithm to obtain embedding matrices from contextualised embedding in an online fashion, i.e. processing the collection in batches and continuously integrating new words or meanings into the present embedding.

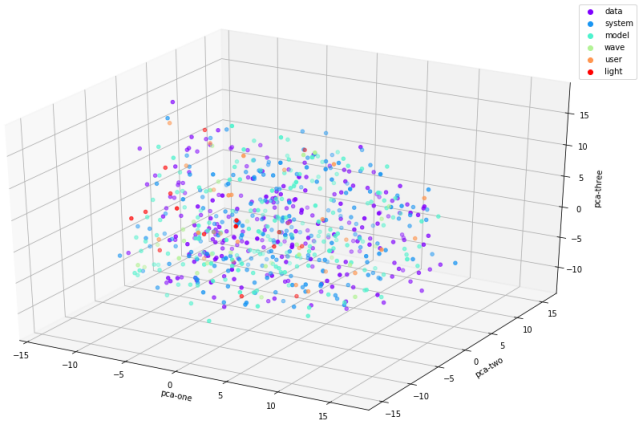


Figure 14: Some more common and less common word vectors, projected to a 3-dimensional space. The embeddings are highly contextualised, to the extent that a significantly different vector is produced for each appearance of the word in the collection.

Secondly, we replace the contextualised embeddings with a pre-trained version of the static embedding GloVe. Specifically, we download the Stanford embedding obtained from the analysis of "Wikipedia 2014 + Gigaword 5", and set the vectors to be 300-dimensional.

In order to better visualise the characteristics of the resulting embedding we have selected a set of words that in our opinion have sufficiently different meanings to end up far away from each other in the vector space. For each such word we print its 10 nearest neighbors according to cosine similarity. The results for DistilBERT with PCA reduction and GloVe are shown in Figures 15 and 16. The gap in the quality of the two embeddings is evident. We also notice that punctuation and seemingly meaningless expressions (e.g. "n", which is probably a variable name) appear to clutter the contextualised embedding space, probably absorbing the meaning of the neighboring words in the sentences they occur in. In order to limit the potential damage of this particular feature of the contextualised embedding on the topic model, we force the vocabulary to only include "intrinsically meaningful" words, which we simplistically identify with nouns. The result can be visualised in Figure 17, which also displays significantly lower quality than GloVe.

```
Visualize word embeddings by using output embedding matrix
word: insurance .. neighbors: ['mimic', 'daily', 'trace', 'cost', 'n', '(', 'determined', 'data', 'glacier', 'even']
word: weather .. neighbors: ['mimic', 'cost', 'n', 'daily', 'trace', 'progress', 'biology', 'negotiation', 'determined', 'even']
word: particles .. neighbors: ['daily', 'mimic', 'arab', 'n', 'trace', '(', 'runoff', '.', 'glacier', '.']
word: religion .. neighbors: ['mimic', 'question', 'interference', 'progress', 'trace', 'daily', '(', 'n', 'cost', 'negotiation']
word: man .. neighbors: ['mimic', 'n', 'cost', '(', 'negotiation', 'daily', 'energies', 'models', 'determined', 'even']
word: love .. neighbors: ['mimic', 'n', 'daily', 'cost', 'progress', 'negotiation', 'trace', '(', 'energies', 'determined']
word: intelligence .. neighbors: ['mimic', 'daily', 'trace', 'energies', 'n', 'arab', 'data', 'pathways', 'question', 'cost']
word: money .. neighbors: ['mimic', 'question', 'daily', 'n', 'cost', 'energies', '(', 'trace', 'interference', 'negotiation']
word: politics .. neighbors: [',', 'question', 'interference', '-', 'mimic', 'although', 'evaluate', 'daily', '-', 'n']
word: health .. neighbors: ['question', 'interference', 'flux', 'evaluate', ')', 'train', 'van', ',', 'chi', 'data']
word: people .. neighbors: ['mimic', 'n', 'daily', 'trace', '(', 'energies', 'negotiation', 'cost', 'determined', 'runoff']
word: family .. neighbors: ['mimic', 'n', '(', 'daily', 'cost', 'negotiation', 'determined', 'energies', 'trace', 'even']
```

Figure 15: Nearest neighbors visualisation of the embedding space for the DistilBERT+PCA embedding.

```
Visualize word embeddings by using output embedding matrix
word: insurance .. neighbors: ['insurance', 'insurers', 'premiums', 'insurer', 'pension', 'insured', 'care', 'savings', 'benefits', 'liability']
word: weather .. neighbors: ['weather', 'inclement', 'rain', 'temperatures', 'rainy', 'conditions', 'storms', 'winter', 'winds', 'rains']
word: particles .. neighbors: ['particles', 'particle', 'molecules', 'electrons', 'photons', 'subatomic', 'atoms', 'protons', 'droplets', 'microscopic']
word: religion .. neighbors: ['religion', 'religions', 'religious', 'christianity', 'beliefs', 'faith', 'belief', 'spirituality', 'catholicism', 'islam']
word: man .. neighbors: ['man', 'woman', 'person', 'boy', 'he', 'men', 'himself', 'one', 'another', 'who']
word: love .. neighbors: ['love', 'loves', 'passion', 'loved', 'romantic', 'lovers', 'lover', 'you', 'me', 'affection']
word: intelligence .. neighbors: ['intelligence', 'cia', 'information', 'security', 'counterterrorism', 'operatives', 'fbi', 'military', 'secret', 'spy']
word: money .. neighbors: ['money', 'funds', 'cash', 'fund', 'donations', 'pay', 'amount', 'paying', 'paid', 'millions']
word: politics .. neighbors: ['politics', 'political', 'politicians', 'religion', 'culture', 'ideology', 'partisan', 'liberal', 'debate', 'social']
word: health .. neighbors: ['health', 'care', 'healthcare', 'education', 'medical', 'hospitals', 'welfare', 'nutrition', 'benefits', 'social']
word: people .. neighbors: ['people', 'others', 'those', 'least', 'many', 'some', 'all', 'them', 'thousands', 'hundreds']
word: family .. neighbors: ['family', 'families', 'relatives', 'father', 'parents', 'mother', 'friends', 'daughter', 'son', 'wife']
```

Figure 16: Nearest neighbors visualisation of the embedding space for the GloVe embedding.

```
Visualize word embeddings by using output embedding matrix
word: insurance .. neighbors: ['values', 'play', 'b', 'avalanche', 'pan', 'contents', 'burning', 'inflation', 'v', 'large']
word: weather .. neighbors: ['young', 'log', 'twist', 'sky', 'conditions', 'clade', 'export', 'status', 'gene', 'st']
word: particles .. neighbors: ['main', 'reaction', 'singapore', 'found', 'comparison', 'spectroscopy', 'effects', 'perspective', 'services', 'r']
word: religion .. neighbors: ['squat', 'utility', 'contact', 'profile', 'scale', 'gene', 'two', 'security', 'plane', 'singular']
word: man .. neighbors: ['tendency', 'survey', 'humidity', 'review', 'path', 'survey', 'range', 'implementation', 'drop', 'moisture']
word: love .. neighbors: ['share', 'cell', 'feeding', 'level', 'drop', 'ice', 'low', 'number', 'low', 'implementation']
word: intelligence .. neighbors: ['low', 'influence', 'gas', 'top', 'image', 'self', 'capture', 'research', 'navigation', 'fuel']
word: money .. neighbors: ['cultivation', 'devices', 'latitude', 'tendency', 'drop', 'activity', 'differential', 'cell', 'longer', 'strength']
word: politics .. neighbors: ['antibody', 'model', 'self', 'wheat', 'concept', 'gene', 'navigation', 'two', 'share', 'strength']
word: health .. neighbors: ['fluorescent', 'aid', 'detection', 'target', 'half', 'term', 'paper', 'study', 'distribution', 'pacific']
word: people .. neighbors: ['tendency', 'survey', 'range', 'carrier', 'generic', 'processing', 'third', 'position', 'intro', 'exchange']
word: family .. neighbors: ['glacier', 'pup', 'research', 'benefit', 'mantle', 'earth', 'one', 'feature', 'magma', 'surface']
```

Figure 17: Nearest neighbors visualisation of the embedding space for the DistilBERT+PCA embedding only including nouns.

Results To run our experiments we resort to the implementation provided with the paper by the authors. The ETM model is realised through a feed-forward neural network, parametrising the posterior distribution and comprising an embedding layer. For our purposes, we keep the model as is, only making changes to the embedding layer when necessary. The result is trained leveraging the automatic differentiation functionality of PyTorch to optimise the Evidence Lower Bound (ELBO) with multiple passes through the data. In order to speed up the inference step the training corpus size is limited to $3,000$ documents, i.e. less than one-sixth of the collection extension. The number of topics for the ETM experiments has been lowered accordingly. For more details on the architecture please refer to our code.

As discussed above, we experiment with three different embedding techniques: namely DistilBERT with "cosine similarity filtering", DistilBERT with PCA reduction and finally GloVe. In contrast to one of the suggested approaches in the paper, we do not update the embedding vectors during training. For each experiment we keep track of the KL divergence, likelihood and perplexity during training and compute topic coherence on a validation set. Moreover, during training we visualise the topics regularly by printing their closest words according to cosine similarity.

In all our experiments the model proved hard to train. Curiously, training on the contextualised embedding raises different issues from those encountered while training on the static word embeddings. We observe almost identical behaviours in the two models obtained from DistilBERT. Specifically, it is possible to notice the KL-divergence term hardly increasing and the reconstruction term not decreasing during training, which signals an inability of the model to evolve its parameters in order to better represent the given data. Figure 35 in the Appendix displays the loss (KL term and reconstruction term) in the last training epochs. Moreover, both the experiments adopting the "semi-contextualised" embedding employ significantly less time in training compared to the GloVe-based ETM: in fact the former take approximately 15 minutes to complete 100 epochs with 50 topics, while the latter takes around 6 hours in the same conditions. Nevertheless, a mild improvement in the topic quality has been observed after the restriction of the vocabulary to nouns. In Figures 18 and 19 we display a set of topics possessing the distinctive features of the final result obtained from the DistilBERT+PCA model, respectively without and with vocabulary restriction.

```
Topic 0: ['processes', '65', 'variation', 'mora', 'specific', 'slow', '(', 'three', 'different']
Topic 1: ['.', 'markets', 'use', 'method', 'field', 'mobile', '%', 'models', 'feature']
Topic 2: ['fleet', 'general', 'changes', 'map', '(', 'data', 'g', 'person', 'rainfall']
Topic 3: ['years', 'reference', 'date', 'quantitative', ',', 'f', 'ag', 'modes', 'architecture']
Topic 4: ['important', 'retention', 'highest', '"', 'weeks', 'reported', 'data', ',', 'external']
Topic 5: ['effect', 'study', 'values', 'induced', 'etc', 'n', 'ago', 'cross', 'model']
Topic 6: ['precipitation', '[SEP]', 'promoted', '.', 'using', 'performance', 'targeting', '-', ')']
Topic 7: ['energy', 'com', 'ann', '[SEP]', ',', 'emi', 'physical', '-', 'combine']
Topic 8: ['build', 'calculation', 'sellers', 'load', 'recently', 'standard', 'sufficient', 'long', ')']
Topic 9: ['acoustic', 'cp', 'required', '0', '38', 'games', 'literature', 'characteristics', 'insights']
Topic 10: ['today', '-', 'future', 'humor', 'natural', 'sets', 'seasons', 'two', 'conclusions']
Topic 11: ['scale', 'fruit', '-', 'system', 'historical', 'cycle', 'used', 'swiss', 'weak']
Topic 12: ['level', 'leading', 'direction', '-', 'used', '-', 'disasters', 'city', 'detected']
Topic 13: ['www', 'bacterial', 'organization', '.', '[UNK]', 'gravity', 'f', 'could', 'shu']
Topic 14: ['para', 'reduced', 'illustrates', 'thus', 'bring', 'vp', 'accuracy', 'following', 'caused']
Topic 15: ['arab', 'diversity', ')', 'defined', 'lead', 'protocol', 'effect', 'eurasian', 'l']
Topic 16: ['design', 'combinations', 'preceded', 'higher', '±', 'fleet', '4', 'health', '"']
Topic 17: [',', ',', '4', '10', ',', 'influence', 'serum', 'detection', 'brightness']
Topic 18: ['ratings', 'performed', 'mini', 'suggests', '(', 'adaptive', 'core', '[SEP]', 'δ']
Topic 19: ['-', 'h', 'implemented', 'wood', 'variety', 'upper', 'consequently', 'look', 'weighted']
Topic 20: [',', 'reflect', 'events', 'core', 'three', 'kinds', 'synthetic', 'aspects', 'well']
Topic 21: ['-', 'ag', 'bladder', '-', 'date', 'elite', '-', 'changes', 'sl']
Topic 22: ['report', 'tracking', 'network', 'model', 'ambitions', 'water', '(', 'dogs', 'clifford']
Topic 23: ['create', 'stopped', 'thus', 'breakthrough', 'models', 'steps', 'phases', '.', 'previously']
Topic 24: ['cluster', 'col', ',', 'thought', 'fungal', 'bio', '900', 'quarter', 'instance']
```

Figure 18: Random subset of Topics produced by the DistilBERT+PCA model.

For what concerns our third experiment, we notice markedly distinct features in both the training behaviour and the final model's results. In particular, the GloVe-based model not only takes significantly longer to process the data, but more importantly its reconstruction term shows an overall decreasing

Figure 19: Random subset of Topics produced by the DistilBERT+PCA model with only nouns in the vocabulary.

trend during inference, proving to be better at modelling the given collection than its contextualised counterparts. Screenshots from the model's final epochs are shown in Figure 36 in the Appendix. In terms of quality of the generated topics the model exhibits two alternative evolution patterns during training, shown in Figures 20 and 21. More explicitly, the model either collapses into a single, narrow and highly coherent topic, or it produces mildly coherent, strongly overlapping, and noisy topics. We hypothesise that the observed outcome can be mainly attributed to two factors: not learning the embedding matrix parameters (as is suggested in the ETM paper) and, most importantly, analysing a relatively small number of documents during training (precisely $3k$ compared to the $\geq 10k$ training points in the original paper).

Figure 20: Random subset of Topics produced by the GloVe model - first pattern.

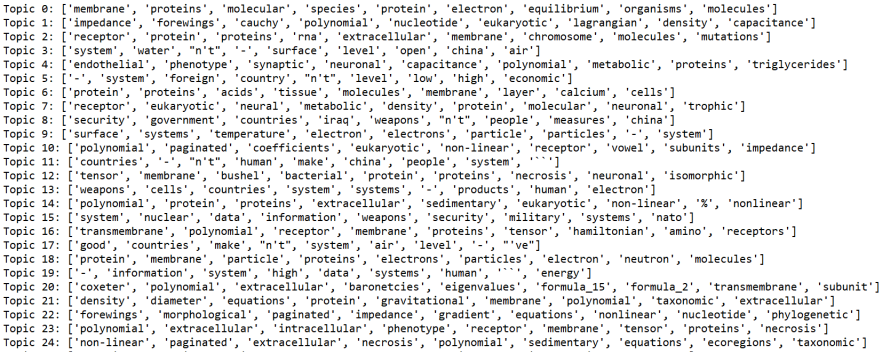


Figure 21: Random subset of Topics produced by the GloVe model - second pattern.

In conclusion, although being a highly promising and modern approach, ETM fails to meet our very first requisite, high performance, irrespective of the embedding provided. However, despite the poor results obtained from ETM, embedding spaces remain a very valuable tool for gaining insights and creating relations beyond topic modelling. As such they remain relevant to the final model implemented.

With ETM our attempts at finding a single topic model that can fit the various needs of the project end. The focus of the remaining part of this report will be on a handcrafted solution building on the models examined so far to best match our requirements.

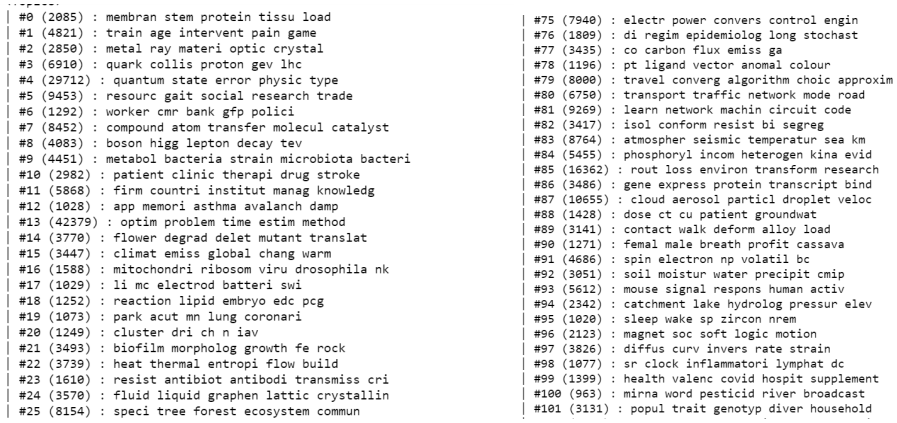
3.7 Final Model

Isolating the topic modelling part of this work and tackling it with existing topic models was not fruitful - not to the extent we had hoped for. However, in our case, topic modelling is not the purpose but a component of the final product. Indeed, the structured data from the ETH research collection has already been integrated in a graph. We can leverage the metadata information available to augment the capabilities of existing topic models and in doing so satisfy more of our requirements. We here describe a purpose built model, *Streaming LDA*, which achieves high performance while allowing online learning and flexibility in the overall number of topics. The training occurs in batches and the topics extracted in each batch will be linked together through two parallel pathways coexisting in the final system. The first of such paths lies in the graph, where by the means of the explicit inclusion of the vocabulary words in it, direct connections between similar topics are naturally drawn. The second one builds upon the embedding space: blending together the features of pre-trained embeddings and topic models we can easily merge multiple topics in a unique vector space. A more detailed description of the model and its versatility is given below.

Streaming LDA - LDA component The foremost requirement of any model implemented for the purpose of the ETH search is high performance. Of the topic models tested and described above, LDA performed the best in this respect and hence forms the base of Streaming LDA. Despite its relative simplicity LDA has proven to be a venerable method, which still forms the baseline and foundation for many other topic models such as ETM. It does not however satisfy any other of the requirements. It is notably not flexible in the number of topics and neither is it capable of lifelong learning. An additional shortcoming of LDA is the fact that is deals poorly with large vocabularies.

Streaming LDA leverages the performance of LDA and addresses directly the aforementioned shortcomings. To simulate continual learning, we run LDA on batches of data. Each batch of publications is used to train a new LDA model and thus new topics are identified for each such batch. Specifically, the dataset of roughly $20,000$ publications is divided into four equal batches. The publications are randomly assigned to each batch, hence ensuring heterogeneity across the batches. By extension, we can assume the the topic distributions to be similar across batches and representative of the dataset. The batch size of $5,000$ was selected on the basis of the publication frequency of ETH and amounts to approximately 6 months worth of publications. Tuning the model with a grid search, armed with the insights gained from previous experiments, has led to 125 topics per batch, with $\alpha = 8 \cdot 10^{-4}$ and $\eta = 2.82 \cdot 10^{-5}$ where α and η are as described in detail in section 3.4. This implies that in every batch 125 topics will be identified. As a sanity check, this can be compared with the results obtained for the baseline experiments presented in section 3.4, where the best performing LDA model trained on the whole corpus of available abstracts contained 450 topics. By simply linearly extrapolating the relationship between number of topics and number of research abstracts, we would observe the optimal number of topics to be a bit more than $4 \times 125 = 500$ for the full corpus. The 50 topics in excess in our selection, is representative of the fact that topics may be repeated across batches, contrary to if LDA were run on the whole corpus in which case new topics should reasonably decrease as more publications get added to the model. This is because it becomes less likely to observe new research topics with a larger observed corpus.

A complete account of the model settings and hyper-parameter values is given in Figure 37 in the Appendix. We include a screenshot from two random subsets of topics extracted by the Streaming-LDA model trained on the first batch (Figure 22). Across the batches there might be overlap in the topics and this is handled, as anticipated above, at the graph and embedding level. We elaborate on this specific step in the next paragraph.



(a) (b)

Figure 22: Topics obtained on the first batch of publications from LDA in its final version.

Note that although the model is fixed in terms of number of topics per batch, there is no limit on the total number of topics covered by the collection. Moreover, as new publications are released, new, potentially unseen topics are inserted in the system and integrated in the graph and the embedding. This allows us to train LDA only once on each document in the research collection, avoiding any further inference steps to update the model parameters. However, these computational benefits come at the cost of memory, as it is now necessary to store 125 topic every $5,000$ documents, which means that the number of topics to store will grow linearly in the size of the research collection.

Streaming LDA - Graph and Embedding component We now proceed to describe the last ingredient in our Streaming-LDA model, namely how the topics are integrated with the the metadata in the graph and the embedding space, which is the most crucial in determining the practicality of the approach.

The research collection is best described as a stream of publications. As mentioned above, based on the data provided to us, approximately $5,000$ publications are released every 6 months. The accuracy of the above estimate is however not relevant for the purposes of this discussion. Applying Streaming-LDA on this ever-growing corpus we obtain one topic model on every such subset of publications. Now the question is: how do we combine these independent models to obtain a seemingly coherent "topic layer" that abstracts and unifies the whole collection?

The first way to join the resulting topics passes through the graph. More explicitly, we expand the structure of the graph as depicted in Figure 23 to include two new node types (representing a topic and a word). We insert the words with highest probability under the topic distribution, covering a total 0.25 of the cumulative distribution, in the graph. Weighted edges are added between the words and the topics they belong to with the weight representing the probability of the word given the topic: $p(\text{word}|\text{topic})$. Similarly, weighted edges are added between the publications and the topics accounting for 80% of the publication-topics distribution as identified by Streaming LDA. The weight on the edge represents the probability mass of the topic in the publication: $p(\text{topic}|\text{publication})$.

The inclusion of the topics' words as nodes in the graph ultimately allows for cross-batch connections to be created. In fact, topics describing similar concepts across batches are linked in the graph through the words they share. This in turn establishes a second-level indirect connection between "similar" documents - i.e., documents involving similar topics - which enables us to identify, given a topic or a word, the set of publications relevant to it, regardless of the input batch they belong to.

Streaming LDA, as described up to this point, does however suffer from some drawbacks. Firstly, the search in the graph database is done with exact matching on the words. Hence if the words of a query are not present in the graph, or in case of incorrect spelling, the query output will be void. Secondly, the issues of automatic topic name inference and topic correlations/hierarchies are not resolved.

One solution which was hinted towards in earlier sections is the use of embedding spaces. Indeed, the intrinsic distance measures that a vector space is enriched with could allow for better query

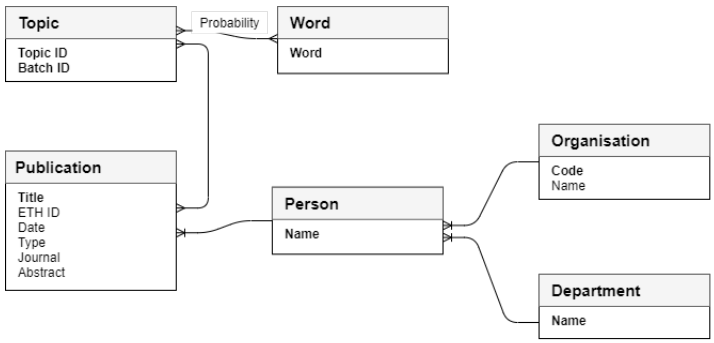


Figure 23: Operational data schema after enrichment.

outputs not only on a topic level but also on a document level through publication similarities for example. Overall embedding spaces could allow for a more flexible querying system, which is a vital characteristic for a system to be put in production. Exploiting the already present connectivity of the graph (in its enriched version) on one hand and a sufficiently robust pre-trained embedding on the other, we are able to associate each node in the research collection graph to an embedding vector. Specifically, the embedding of each topic is obtained through a convex combination of the embedding of its most relevant words, and likewise the embedding of a document is computed as a convex combination of the topics it contains. 91% of the words obtained from the topic representations are successfully matched by existing words in the embedding space. The embedding of the remaining nodes in the graph follows the same logic. More details on how the node embedding vectors are obtained is given in subsection 4.2. Notice that following the definition of topic embedding "similar" topics will end up close in the embedding space, irrespective of their batch. Likewise, conceptually similar documents will be represented by geometrically similar vectors. Thus, as anticipated above, the embedding provides a valid alternative way to overcome the limitations of Streaming LDA, while preserving its qualities. Moreover, exploiting the intrinsic features of vector spaces, we can again solve, as it was intended for ETM, our remaining requirements of topic name inference and topic hierarchies.

There is one final detail regarding the final model which is worth mentioning before proceeding to the description of the querying component of the system. The pre-processing required for the two facets of Streaming LDA is currently different. On the graph side, applying normalisation in the form of token stemming and lemmatisation leads to an increased matching rate on the words. This step however may change the form of certain words such that they no longer are recognisable by language representation models such as GloVe. Consequently, the input to the LDA model used to enrich the embedding space is not normalised. This choice doubles the training time on each batch, a price that we believe worth paying for the resulting benefits.

4 Querying

The output from Streaming LDA is used to enrich the graph space and the embedding space. As such there are two parallel streams of information which can be queried, one on the graph side and the other on the embeddings side. The graph database used, Neo4j, can be queried using Cypher queries while the queries in the embedding space are performed through a specified similarity function.

4.1 Graph side

One of the targets of this work is to equip the ETH Search system with the notion of "experts" in different fields. As shown in Figure 1 in the introduction, currently searching for people related to Climate Change at ETH returns no results, yet a query of the sort is very relevant to other researchers & collaborators, possible PhD applicants, future students and to the professors themselves. We consider researchers to be experts in a field if they have authored numerous publications which relate strongly to the said field. In addition, we do not allow the definition of expertise to generalise to "too

broad" fields. On the graph side, we implement this concept through a direct connection between authors and topics. Specifically, the problem is formulated as follows: given a query consisting of one or more words which identifies a field, find its most prominent authors.

Enriching the Graph The solution in our system consists of a Cypher query, which we now proceed to illustrate. In order to properly order and filter the query output new weights are inserted on the existing graph edges. First, we include weights between the authors and publications to highlight more recent publications:

$$r_p = 1 - \frac{\texttt{publication year}}{2020 - 1930},$$

where r_p encodes the recency of the publication and the denominator indicates the time span of the research collection. We will later use the recency factor to scale the "expertise" of a professor on a topic. In this way we encourage the system to recommend authors that are currently publishing in the field, as we consider this information to be more valuable for the prototypical user. Two additional weights are created to penalise topics and publications which lack specificity. If a topic's probability distribution is nearly uniform over the vocabulary, or a topic is shared by a considerable amount of publications, it may indicate that it is a very general one. In contrast, a topic identified by few prominent words and sparsely connected with the publications better describes the true nature of the publication. The above concepts translate into the following expressions:

$$tsw = \texttt{count}(w_t)$$
$$tsp = \texttt{count}(p_t),$$

where w_t identifies the set of words linked to the topic t and p_t refers to the set of publications associated with the topic t. These elements will together measure the conceptual extension of the topic, which acts as a discount factor on the expertise value. Finally, authors having more publications linked to a topic are more likely to be an expert on it. We synthesise this concept in the number below:

$$a_t = \begin{cases} 1, & \text{if } \texttt{count}(p_{a,t}) \geq 10 \\ \left(\dfrac{\texttt{count}(p_{a,t})}{10}\right)^{0.75} & \text{otherwise,} \end{cases}$$

where $p_{a,t}$ identifies the set of publications authored by a within topic t. Note that the number of publications relative to a topic is processed so as to mitigate the effect of unnaturally high values, and only promote the authors with more than 10 publications on the topic. The probabilities resulting from LDA ($p_t = p(\text{topic}|\text{publication})$) combined with the aforementioned custom factors define the level of expertise of an author on a topic, which is directly integrated in the graph structure as the weight of a new relationship `EXPERT_ON` between authors and topics.

$$\texttt{EXPERT\_ON weight} = \texttt{AVG}\left(r_p \cdot p_t\right) \times \left(\frac{9}{tsw}\right)^{0.75} \times \left(\frac{63}{tsp}\right) \times a_t$$

The values on the numerator which weigh the inverse topic specificity both in terms of words and publications represent the minimum number of words a topic contains and the minimum number of publications a topic refers to, as observed in the data present in the graph. This choice ensures an inverse relationship between the `EXPERT_ON` weight and the generality of topics, i.e. a more general topic represented by a larger number of words and present in a larger number of publications will receive lower expert scores.

The thresholds and constants used in the above formulas are rather arbitrary and experiment-based. Multiple internal sources at ETH agree that a more truthful estimate of the expertise of an author relative to a given topic should take into account the existing differences between departments and research groups. Thus, in a more sophisticated reformulation of the above score the thresholds and constants could vary depending on the author's department or research group.

At this stage the graph contains the information available from the metadata as described in subsection 2.3, the output from Streaming LDA and meaningful relationships between all of the above. This distils to a highly queryable, fast and efficient database from which it is possible to extract meaningful insights with regards to publications, ETH staff, interdepartmental co-operations and much more. In the next section we put everything together and illustrate the unrolling of the querying pipeline on the graph side on an example query.

Querying In order to demonstrate the potential of the enriched research collection database as depicted above, we consider again the scenario of Figure 1. As mentioned earlier, Streaming LDA on the graph side performed best with normalisation and the same normalisation has to be applied to queries relating outputs from LDA, such as identifying experts in a field or papers relevant to a topic. This is portrayed in Figure 24a where the query "Climate Change" becomes a list of normalised words `['climat', 'chang']`.

The intrinsic properties of graph databases together with the additional layers of abstraction embedded into the enriched version of the graph make the task of extracting the information relevant to the input query relatively easy and efficient. In fact, leveraging the shortest path algorithms implemented in Neo4j we are able to return a list of authors ordered by their expertise score, given the words in the input as starting nodes. The code and result for the chosen example query are shown in Figure 24a and Figure 24b respectively. Note that the computational cost for the query is kept low by performing the enrichment step offline. This choice shifts most of the computational burden on the data insertion step, which is assumed to occur significantly less frequently than the querying.

```
WITH ['climat', 'chang']
as words
MATCH (w:Word)-[r1:IS_IN]-(t:Topic)
WHERE w.name in words
WITH t, size(words) as inputCnt, count(DISTINCT w) as cnt, SUM(r1.weight) as s
WHERE cnt = inputCnt
WITH  t, s
MATCH (t:Topic)-[r3:EXPERT_ON] - (p:Person)-[r2:WORKS_IN]-(d:Department)
WITH p, SUM(r3.score*s) as s2, d
RETURN p.name, s2, d.name
ORDER BY s2 DESC
```

(a) Cypher query to obtain ETH faculty whose research expertise is related to Climate Change.

p.name	s2	d.name
"Lohmann, Ulrike"	0.010928467789625685	"Environmental Systems Science"
"Knutti, Reto"	0.009681374250352134	"Environmental Systems Science"
"Gruber, Nicolas"	0.009511072975518802	"Environmental Systems Science"
"Peter, Thomas"	0.007843464338036742	"Environmental Systems Science"
"Buchmann, Nina"	0.00779752948292354	"Environmental Systems Science"

(b) Top 5 ETH researchers matched from a query on climate change

Figure 24: Querying the graph to obtain the most relevant ETH staff with Climate Change as their area of expertise.

In the example Cypher query `r3.score` is equivalent to the `EXPERT_ON` weight as defined above. To achieve a more robust output from this query, additional weight is given to researchers whose publications match words with high $p(\text{word}|\text{topic})$. This is integrated in `s = SUM(r1.weight)`.

The provided example only covers the specific case of experts retrieval. However, the system developed allows us to extract all the information relevant or related to the input query. For instance, given the "Climate change" input query, it is now possible to retrieve the publications most relevant to it ordered by recency, or the most prominent research groups in the field, perhaps ordered by the number of experts they hold. For a different input query, say specifying an existing professor, we can efficiently retrieve their most common collaborators, the most recent publications, as well as all the metadata information such as related research groups and the department they belong to. In a nutshell, the structure that has been developed around the raw research collection data permits us to access the underlying data efficiently, to query it to match arbitrarily complex patterns and in so doing produce insights into the status of research at ETH.

The above examples all assume the input query to be in a rather rigid format, directly matching the node identifiers in the graph. While the embedding side of the querying system partially overcomes these limitations, it is safer to assume that a more sophisticated pre-processing system is applied to the input query and that its output is fed into our system. The problem of correctly identifying the entities participating in a natural language expression, known as Named Entity Recognition (NER), has been addressed already multiple times in the past, and today various solutions are available, some of them implemented in common NLP libraries such as NLTK and SPACY. In the interest of time, we have decided to leave the integration of such a system into our solution to future work. For the time being, in terms of authors, the pre-processing is such that the query becomes `['surname', 'name']` which is the way it is included in the graph. This notably may cause difficulties for compound surnames for example, where partial matching would be required.

As anticipated in earlier sections, the input queries are submitted to two parallel streams of computation designed to complement each other in the retrieval of information from the collection. We now proceed to present the second of these two components in more detail, which relies on the definition of graph node embeddings.

Embedding topics Having obtained word vectors from a pretrained embedding (GloVe), we can use the weights encoding the topic distribution over the input vocabulary to obtain topic embeddings. Specifically, the topic vectors are realised as a weighted combination of the word vectors, after the weights have been normalised to 1. In fact, each topic is computed only from its most descriptive words adding up to 25% of the probability mass. Similar topics in terms of cosine similarity can now be visualised as for example in Figure 25 below, where the two most similar topics related to a given topic about molecular biology are displayed. Note that there are only 125 topics per batch, showing that similarities can be identified across batches as is the case here, indicating the ability of Streaming-LDA to integrate topics across batches. Moreover, in Figure 38 in the Appendix section a random subset of topics is visualised in a lower-dimensional projection of the embedding space.

```
Topic 11
genotypes peptides cas proteomics food peptide genomes clock sequence complex crispr domain molecular tolerant sensitivity d
ynamics encoding contained encoded object behaviors prevalence associated cooperative gdap people serves plasticity roles mu
ltiple targeted driver projects strains cortical toc fine circuit wild corn sequences phylogenies together derived factor an
cestors deployment frames naturalistic stable eukaryotes revealed recurrent lineage environmental loci rapidly could eight w
hether vulgaris end frequency extended cdic microbial
----------
Topic 193
genes gene expression genome metabolic sequencing regulation coding functional protein pathogen transcription mutations path
ways dna biological expressed suggesting alterations evolutionary promoter single transcriptome stress maintenance breeding
phenotypic plants known syndrome locus regulatory including lineage mediated analysis evolved eukaryotic transcript non prev
iously epigenetic analyses pattern regulator innate rrna changes levels involved markers mrna type fibres chromatin folding
resource effector

Topic 255
rna cellular regulatory cell human acid phosphorylation drug levels molecules mammalian metabolism ago cells substrates targ
eted endogenous proteins effector lines embryonic transcriptional pathogen revealed targeting identified maintenance adaptat
ion metabolic efficacy content drugs transcription components transgenic complex secreted tuberculosis antigen act induced a
nti understood perceptions cycle either mechanisms candidates expression canonical o_ switch findings small sirna show gener
ated experiments unlike novo ifn generate newly phosphate maintained enable processes involvement wild development combinato
rial gram
```

Figure 25: Topic similarity example. The two most similar topics to Topic 11 in this case are shown.

4.2 Embedding side

Embedding documents Once topic vectors are available, it is possible to obtain embedding vectors for the publication nodes, leveraging the representation of a document in tuples of topics and weights, as identified by Streaming LDA. The weight corresponds to the probability mass of the topic in the given document. As mentioned above, 80% of the most prominent topics in each publication are included in the embedding space and hence in similar fashion to before, the weights are normalised to 1 and the documents are embedded as a convex combination of the topic vectors. The similarity of pairs of documents can again be visualised by measuring the cosine similarity between the document vectors, an example of which is shown in Figure 26.

Embedding authors Finally, author nodes are incorporated in the embedding space. Note that it is in principle possible to obtain embedding vectors also for the research group and department nodes, adopting a logic similar to the one applied in the case that we are about to discuss. We let the author embeddings also be defined as a convex combination of their respective publications. This task is

```
Document 5
As the strategic rivalry between the US and China intensifies, militarized crises are becoming more likely and a major militar
y conflict is no longer as remote as it once seemed. The far-reaching modernization of its armed forces has already led China
to embrace a more sanguine view of how such a conflict might play out. Meanwhile, the United States is struggling to formulate
a coherent response to a potential Chinese attempt to recast the regional order by force. Although war remains unlikely, the n
eed to get real about the possibility is now more urgent than at any point in recent decades.
----------
3 most similar documents in the collection

Document 31
The relevance of nuclear weapons in world affairs is increasing, not decreasing. All nuclear powers modernize their arsenals.
This may result in destabilizing effects on nuclear deterrence constellations. At the same time, the discrepancy between the i
mportance of arms control as a necessary supplement to nuclear deterrence on the one hand and its actual, limited role in inte
rnational affairs on the other hand is constantly growing. In order to avoid future nuclear wars and to create strategic stabi
lity, a renaissance of arms control is urgently needed.

Document 105
Strategic Trends 2019 offers a concise analysis of major developments in world affairs, with a focus on international securit
y. In the first chapter, Jack Thompson considers the consequences of the Trump administration's new approach to trade policy.
In his view, the United States is powerful enough to extract trade concessions from all of its trading partners, and there may
be some short-term advantages in following such a course of action. In the second chapter, Michael Haas examines the shift bet
ween the West and non-Western states in the field of defense technologies. He argues that Western policymakers should act on s
everal fronts to slow the process, while also adapting to a world in which they no longer enjoy substantial military-technolog
ical superiority. In chapter three, Jeronim Perović considers the emergence of the Eurasian Economic Union (EAEU), a surprisin
gly robust multilateral organization of post-Soviet states, which is not a Russian puppet, and which cooperates in economic, p
olitical, and military matters. Finally, in chapter four, Lisa Watanabe looks as Russia's re-emergence as a power broker in th
e Middle East and North Africa, with a focus on countries of particular interest to Europe when it comes to security issues, e
conomic ties, and immigration.

Document 86
Does nuclear proliferation have stabilizing or destabilizing effects? This question is fascinating for scholars of the nuclear
age and highly consequential for practical policy issues. For in order to debate the merits of particular policy choices—such
as preventive military strikes against nuclear facilities, grand bargains with potential proliferators, or complete nuclear di
sarmament—we first need to understand how the spread of nuclear weapons impacts regional and global security. To enhance our u
nderstanding of this crucial issue, this chapter engages the empirical literature on the consequences of proliferation, focusi
ng on how the pursuit and acquisition of nuclear weapons by additional states have influenced international stability. It also
explores whether some states have been more affected than others, and what measures these states have taken to prevent prolife
ration or to mitigate its negative consequences.
```

Figure 26: Document similarity example. The abstracts corresponding to the three most similar documents to one related to US-China military relations are shown.

less straight-forward than the previous two however, as we do not have access to a normalised set of weights linking each author to their publications. We thus need to rely on a custom-made scoring function describing the relevance of a publication to a given author. Three factors are taken into consideration when computing this score: publication type, publishing date and number of authors. The publication types are attributed different scores, with conference papers scoring the highest (10) and review articles the lowest (7). This particular choice embodies the assumption that conference papers, for instance, better characterize the work and interest of a professor than their review articles. Regarding the date of the publication, more recent publications are given a higher score according to the exponential of the difference to the global mean of the publication dates. As for the number of authors, publications with many authors are penalised according to the purely domain-knowledge based principle that publications with multiple authors might not be as representative for each author's expertise. This penalisation is done by taking the number of authors to the power of a coefficient γ, here chosen to be 0.5. The score is then computed as the product of these three values. For the author embedding we threshold the number of publications to be considered for the embedding of an author to be max 20, including only the publications with the largest scores if their cardinality exceed this number. This last filtering step prevents noisy records to enter the estimate. The author embedding is then computed as a linear combination of the selected publications, weighted by the normalised scores. We note that this particular choice of score function is highly heuristical. There are many other factors that might determine the order of relevance in a publication portfolio, which furthermore might be subjective, change over time and depend on the research area. Nevertheless, we present an example of the resulting embedding space also for authors in Figure 27 below. As can be inferred from this example, there is indeed room for improvement and experimentation for this scoring function. More specifically, we notice that while the second and third suggested authors appear to be substantially similar –in terms of the focus of their work– to the submitted one, the very first option emerges in net contrast with the rest. We hypothesise this odd behaviour to be partially attributable to a disproportionate sensibility of the overall system to the extension of each author's publication history. Further investigation is still needed to confirm the claim. In Figure 39 in the Appendix an additional visualisation of the embedding author vectors is given.

```
Author 19795
Name : Ghaffari, Mohsen
Some publications:
- A Massively Parallel Algorithm for Minimum Weight Vertex Cover
- A simple parallel and distributed sampling technique: Local glauber dynamics
- A tight analysis of the parallel undecided-state dynamics with two colors
- Derandomizing distributed algorithms with small messages: Spanners and dominating set
- Distributed Algorithms for Low Stretch Spanning Trees
- Distributed MST and broadcast with fewer messages, and faster gossiping
- Distributed set cover approximation: Primal-dual with optimal locality
- Faster Algorithms for Edge Connectivity via Random 2-Out Contractions
- Improved distributed degree splitting and edge coloring
- Leader Election in Unreliable Radio Networks
----------
Author 57378
Name : Stoop, Norbert
Some publications:
- Fluid membrane vesicles in confinement
 - Simulating thin sheets: Buckling, wrinkling, folding and growth
 - Subdivision shell elements with anisotropic growth

Author 33940
Name : Leucci, Stefano
Some publications:
- An improved algorithm for computing all the best swap edges of a tree spanner
 - Effcient oracles and routing schemes for replacement paths
 - On the PSPACE-completeness of Peg Duotaire and other Peg-Jumping Games
 - On the complexity of two dots for narrow boards and few colors
 - Optimal Sorting with Persistent Comparison Errors

Author 8224
Name : Böckenhauer, Hans-Joachim
Some publications:
- Constructing Randomized Online Algorithms from Algorithms with Advice
 - On the Power of Advice and Randomization for the Disjoint Path Allocation Problem
 - On the advice complexity of the knapsack problem
 - Online algorithms with advice
 - Reoptimization of the Shortest Common Superstring Problem

Author 28200
Name : Kahles, André
Some publications:
- A nearly optimal algorithm for the geodesic voronoi diagram of points in a simple polygon
```

Figure 27: Author similarity example. The author used in this example is Prof. Ghaffari whose research focus is on distributed algorithms, parallel algorithms, network algorithms, and randomized algorithms. The research field of the most similar author identified in this example does not seem to match that of our query, however the next few (only 3 additional shown here due to space constraints) seem to have similar research interests.

Embedding queries With the exception of organisations and departments, the above steps have served to embed the graph structure into a more robust queriable space thanks to word embeddings. In other words a multidimensional vectorial space has been created in which topics, authors and publications coexist. As such it is now possible to translate a query into a vector as well and search the embedding space for the most suitable answer. The query is first broken down into a list of its words (tokens) and these are matched to their GloVe embedding counterparts. The resulting vector is taken as the mean of the individual embedding vectors and the output consists of identifying the closest vectorial representations in terms of cosine similarity for the topics, publications and authors. To remain consistent with the above discussion we exemplify the potential of the embedding approach again using the "Climate change" query. In Figure 28a and Figure 28b the result of the search for topics and documents is visualised. For more details on the how the embedding space was enriched or how the querying is performed the reader is referred to the code.

```
Query: climate change
----------
10 most similar documents in the collection

Document 6540
Determining the time of emergence of climates altered from their natural state by anthropogenic influences can help inform the
development of adaptation and mitigation strategies to climate change. Previous studies have examined the time of emergence of
climate averages. However, at the global scale, the emergence of changes in extreme events, which have the greatest societal im
pacts, has not been investigated before. Based on state-of-the-art climate models, we show that temperature extremes generally
emerge slightly later from their quasi-natural climate state than seasonal means, due to greater variability in extremes. Never
theless, according to model evidence, both hot and cold extremes have already emerged across many areas. Remarkably, even preci
pitation extremes that have very large variability are projected to emerge in the coming decades in Northern Hemisphere winters
associated with a wettening trend. Based on our findings we expect local temperature and precipitation extremes to already diff
er significantly from their previous quasi-natural state at many locations or to do so in the near future. Our findings have im
plications for climate impacts and detection and attribution studies assessing observed changes in regional climate extremes by
showing whether they will likely find a fingerprint of anthropogenic climate change.

Document 11599
Based on high-resolution models, we investigate the change in climate extremes and impact-relevant indicators over Europe under
different levels of global warming. We specifically assess the robustness of the changes and the benefits of limiting warming t
o 1.5°C instead of 2°C. Compared to 1.5°C world, a further 0.5°C warming results in a robust change of minimum summer temperatu
re indices (mean, Tn10p, and Tn900p) over more than 70% of Europe. Robust changes (more than 0.5°C) in maximum temperature affe
ct smaller areas (usually less than 20%). There is a substantial nonlinear change of fixed-threshold indices, with more than 6
0% increase of the number of tropical nights over southern Europe and more than 50% decrease in the number of frost days over c
entral Europe. The change in mean precipitation due to 0.5°C warming is mostly nonsignificant at the grid point level, but, loc
ally, it is accompanied by a more marked change in extreme rainfall.
```

(a) Most relevant publications to the query as measured by cosine similarity.

```
Query: climate change
----------
10 most similar topics

Research topics 103
ice climate land change carbon global water freezing nucleation regional

Research topics 230
climate precipitation psma europe extremes cmip projections sn temperature µg

Research topics 497
climate change cost changes regions strategies future industry sectors regimes

Research topics 178
species climate biodiversity mitigation ecological vegetation effects policy electricity policies

Research topics 261
precipitation water climate warming aerosol temperature heat impacts models land

Research topics 203
air climate seismic temperature annual surface rainfall rock site northern

Research topics 334
lake carbon ocean climate flux land surface vegetation ecosystem productivity
```

(b) Most relevant topics to the query, as measure by cosine similarity.

Figure 28: Querying the embedding space to obtain the most relevant ETH publications and the topics on Climate Change.

In summary, having estabilished a mapping between the metadata information and its vectorised form unlocks a new paradigm for search. In fact, while the graph excels at matching complicated relational patterns and hence extracting conceptually elaborate information, the embedding space enjoys more flexibility in term matching and similarity measures, allowing the efficient and robust retrieval of the relevant, albeit unsophisticated knowledge present in the research collection.

5 Future Work

We have thus far created a model which acts as a proof of concept for future directions to consider when redesigning the ETH Search. Streaming LDA meets a number of the requirements set out but nonetheless, when designing it, multiple points of future improvement were identified in our experiments and literature which could render it faster, more accurate and more robust.

The quality of any match originating in the graph as well as the definition of the topics in the embedding space, are dependent on the output of LDA. One notable example where this may prove to be a limitation is when thinking about papers on a specific topic within a domain such as Machine Learning. This bigram, "Machine_Learning", is not commonplace in such publications which will use much more precise terms. In contrast, a Social Science paper implementing regression for example, may be loaded with the term. As such, a social scientist using Machine Learning as a tool, may be identified as an expert in the field, while a Professor in Computer Science specialising in a subfield of Machine Learning, may not be identified by the model at all.

This naturally leads to two possible lines of development. Firstly, improving on the topic model itself, either by better tuning LDA or switching it with a more suitable model and secondly, by extending the use of embedding spaces. The first point was to a great extent the focus of a major part of the project and within the threefold system of topic model - graph - embedding space, LDA confidently outperformed the other models. The output of LDA could nonetheless be augmented through Professor website information or even the ETH course catalogue. This could be used for example to name topics, validate them, cluster them and add words that may be relevant to them. The bigram "Machine_Learning", could for example be added retroactively in cases such as the example above. Embedding spaces on the other hand are very promising and have reshaped NLP tasks in general. Currently the embedding pipeline implemented is quite rigid, with the use of GloVe and the words extracted from LDA to perform the embedding. A natural extension would be to leverage the richness of the research collection, the connections in the graph and initialise the embeddings with GloVe, but use architectures such as Graph Neural Networks (GNNs) to create a more representative embedding. This approach touches upon the general realm of representation learning which could be introduced in the current model modularly, by using it to encode the structured data into the embedding space or to overhaul the whole model as described with the GNNs to create a more purpose-built embedding. As a reminder, the structured data such as the publications, are currently added into the embedding space as convex combinations of their nearest embedded connections in the graph, namely their topics in this case.

There are also numerous less drastic changes and inclusions that can be made to improve the outcome of queries. For example, named-entity recognition could be added in query time to identify the important parts of a query and by extension pre-process the query accordingly or prioritise either the embedding space or the graph space to obtain a match. Additionally, in the pre-processing performed on the corpus, bigrams and trigrams could be added such that words oftentimes occurring in succession as "Data_Science" and "Machine_Learning" are better identified in the topics.

Finally, further static information could be included to assist in the ordering of the publications as used in the score functions for query outputs. These could include the citation count, first authorship and journal ranking, although one might think of many others (which may or may not be accessible in the provided dataset).

6 Summary

In this report we have described a self-contained system for the improvement of the ETH search. It leverages the performance of LDA, the strengths of embedding spaces and the optimality of graphs in a holistic model - Streaming LDA. Streaming LDA extracts information from unstructured data in batches and includes them in a graph. The graph contains information from the structured data in the research collection as well as the unstructured data from previous batches. Within the graph relationships are created which allow for connections and insights to be gained across all levels of ETH, ranging from author collaborations and expertise, to department publication numbers. Finally the graph is encoded into an embedding space which allows to perform more indirect querying, which at times proves somewhat less accurate, but overall permits the discovery of more varied relations and facilitates querying.

The final model is the culmination of insights gained from various topic modelling experiments. HDP for example was useful in helping identify the order of number of topics while ETM in gaining insight into the utility of embedding spaces. Overall Streaming LDA qualitatively and to some extent quantitatively uncovered a potential direction for the future implementation and use of the ETH search, satisfying our requirements of high performance, online learning and flexibility in the number of topics.

Acknowledgements

We thank Paul Cross, Christine Khammash, Tarun Chadha and Professor Zhang for the constant feedback and support provided throughout the project. Furthermore, we thank the Institutional Research group and Rao Xi for the help offered in the data retrieval.

References

[1] F. Gong, Y. Ma, W. Gong, X. Li, C. Li, and X. Yuan, "Neo4j graph database realizes efficient storage performance of oilfield ontology," *PLOS ONE*, vol. 13, no. 11, pp. 1–16, 11 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0207595

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407.

[3] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, ser. NIPS'09. Red Hook, NY, USA: Curran Associates Inc., 2009, p. 288–296.

[4] A. Agrawal, W. Fu, and T. Menzies, "What is wrong with topic modeling? and how to fix it using search-based software engineering," *Information and Software Technology*, vol. 98, pp. 74 – 88, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584917300861

[5] S. Koltcov, V. Ignatenko, and O. Koltsova, "Estimating topic modeling performance with sharma–mittal entropy," *Entropy*, vol. 21, no. 7, 2019. [Online]. Available: https://www.mdpi.com/1099-4300/21/7/660

[6] L. Xing, M. J. Paul, and G. Carenini, "Evaluating topic quality with posterior variability," *CoRR*, vol. abs/1909.03524, 2019. [Online]. Available: http://arxiv.org/abs/1909.03524

[7] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, ser. WSDM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 399–408. [Online]. Available: https://doi.org/10.1145/2684822.2685324

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 993–1022, Mar. 2003.

[9] D. M. Blei and J. D. Lafferty, "Correlated topic models," in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, ser. NIPS'05. Cambridge, MA, USA: MIT Press, 2005, p. 147–154.

[10] W. Li and A. McCallum, "Pachinko allocation: Dag-structured mixture models of topic correlations," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 577–584. [Online]. Available: https://doi.org/10.1145/1143844.1143917

[11] T. G. Mark Steyvers, *Probabilistic Topic Models*.

[12] Y. Teh, M. Jordan, M. Beal, and D. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, pp. 1566–1581, 01 2006.

[13] C. Wang, J. Paisley, and D. Blei, "Online variational inference for the hierarchical dirichlet process," *Journal of Machine Learning Research - Proceedings Track*, vol. 15, pp. 752–760, 01 2011.

[14] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," 2019.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[19] K. Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 55–65. [Online]. Available: https://www.aclweb.org/anthology/D19-1006

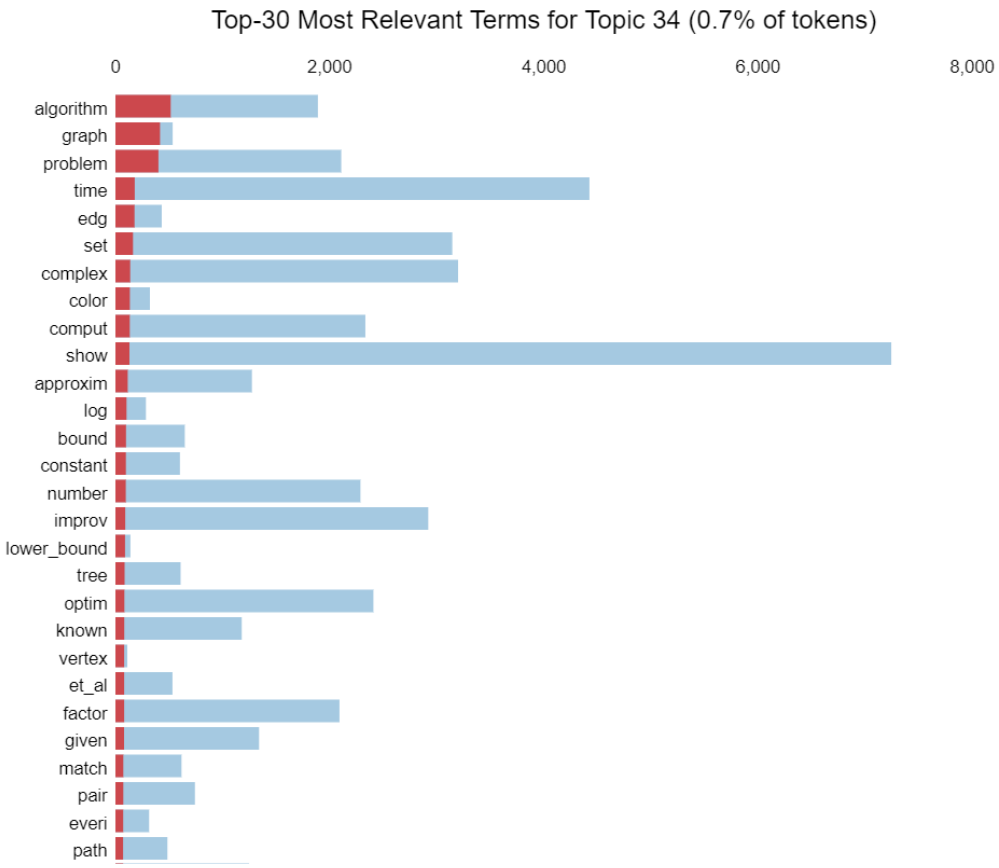


Figure 29: LDA1 model Topic 34

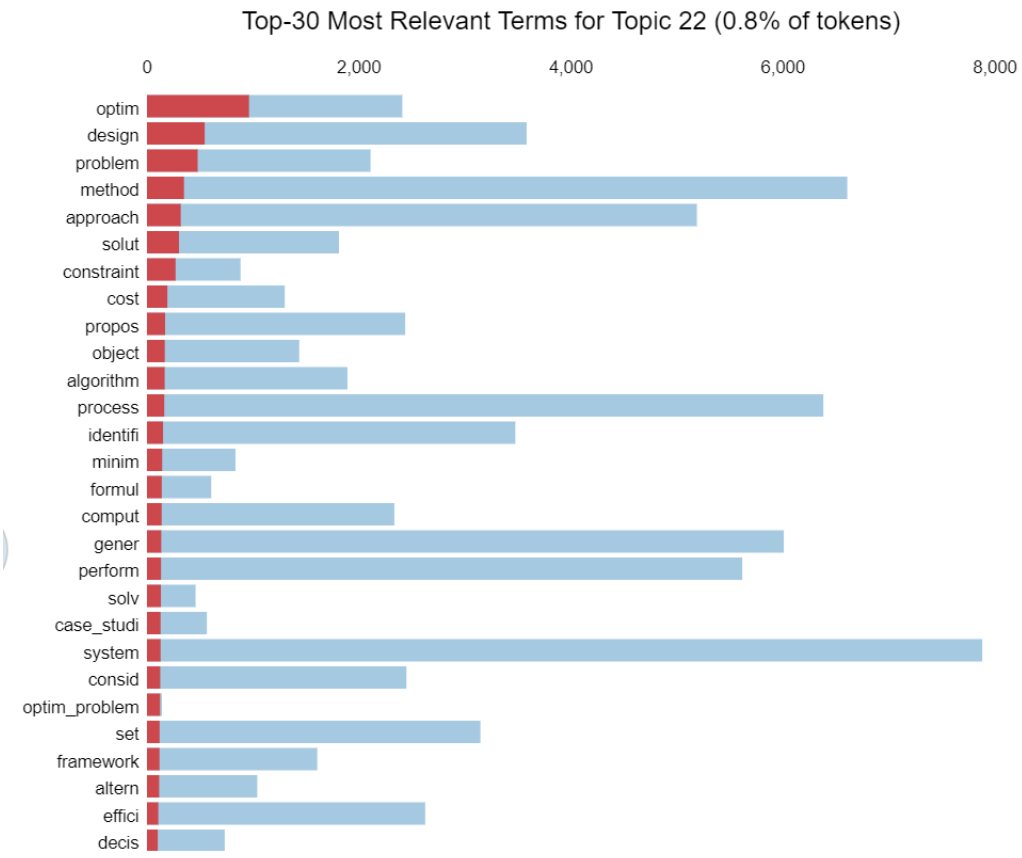


Figure 30: LDA1 model Topic 22

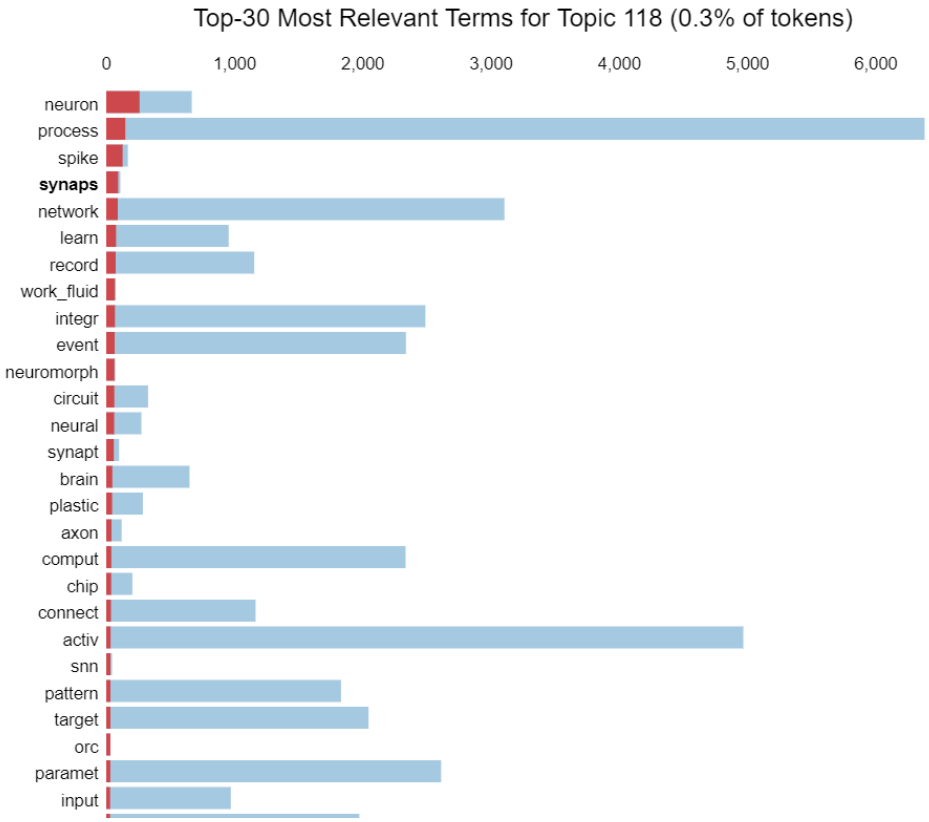


Figure 31: LDA1 model Topic 118

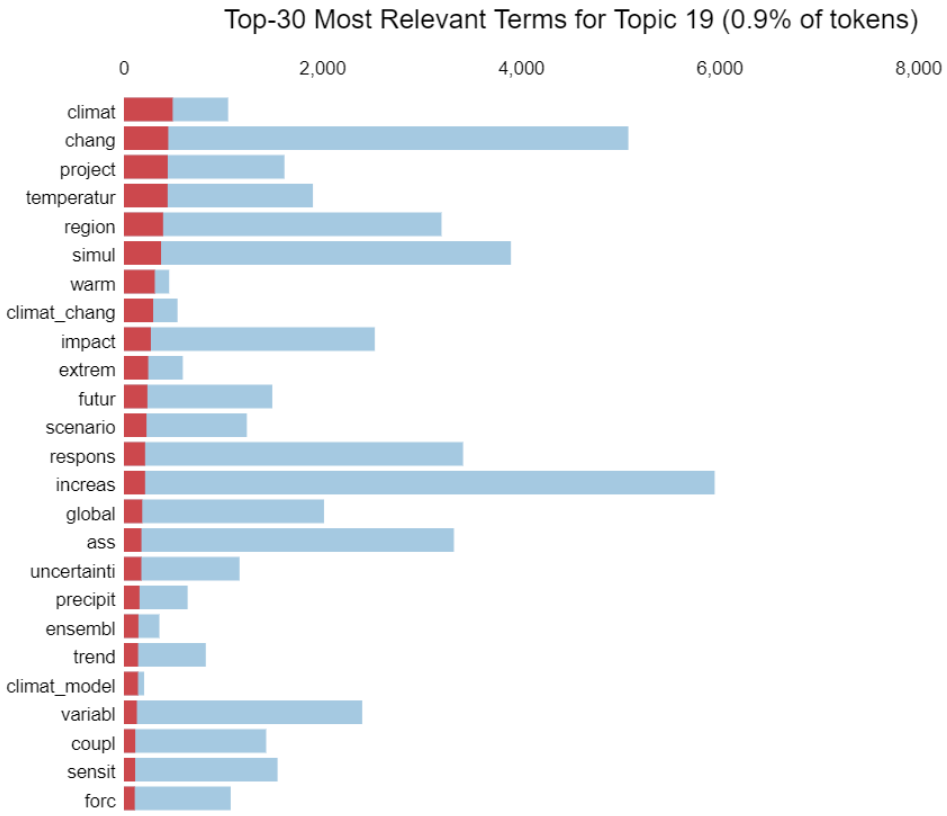


Figure 32: LDA1 model Topic 19

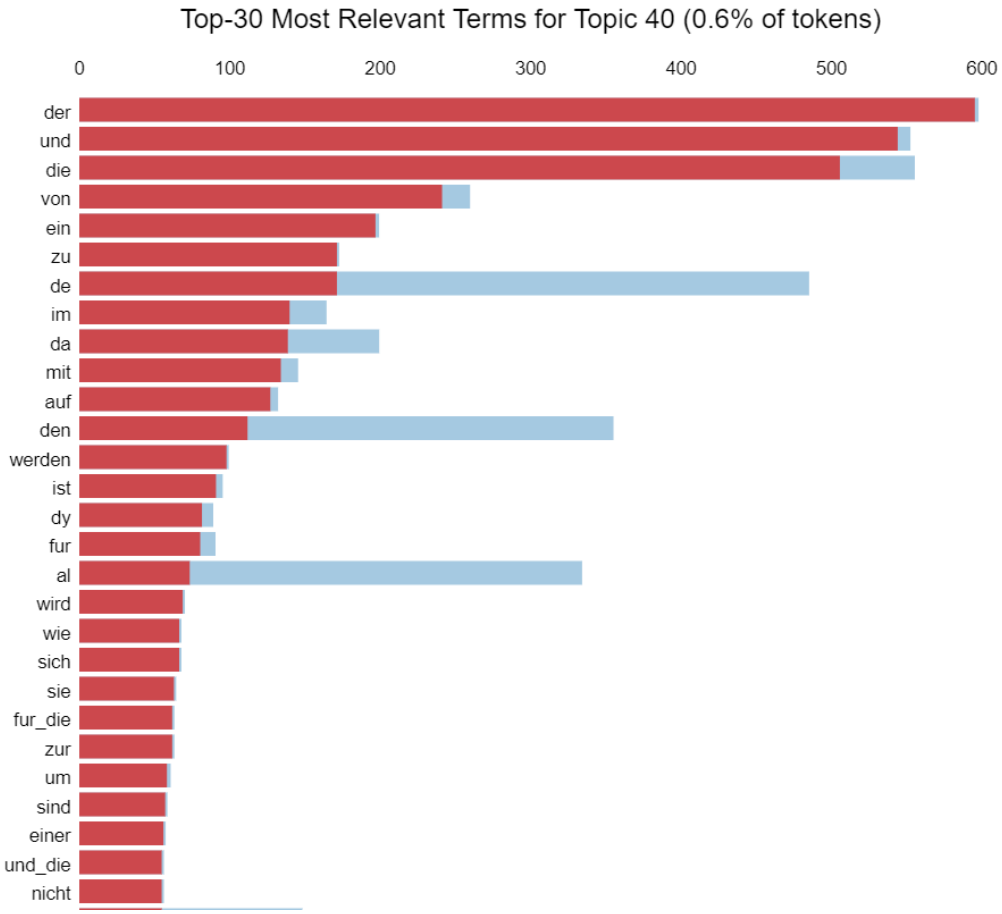


Figure 33: LDA1 model Topic 40

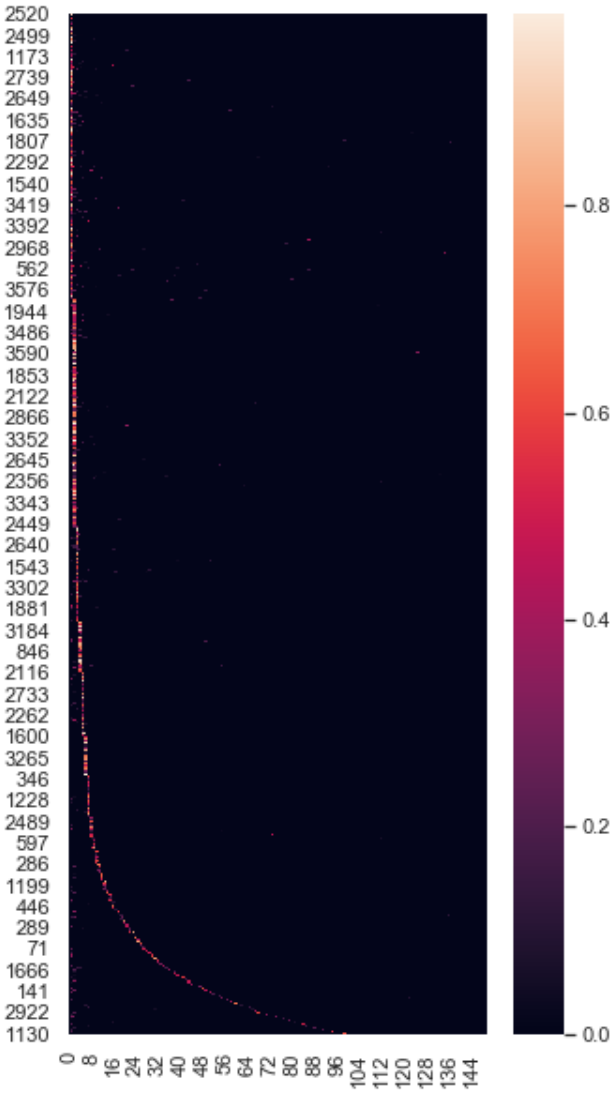


Figure 34: Documents × First Topic Matrix from the Online HDP model

```
Epoch----->97 .. LR: 0.05 .. KL_theta: 0.07 .. Rec_loss: 1515.59 .. NELBO: 1515.66
Eval Doc Completion PPL: 26901.3
Training finished.
Epoch: 98 .. batch: 2/5 .. LR: 0.05 .. KL_theta: 0.08 .. Rec_loss: 1453.27 .. NELBO: 1453.35
Epoch: 98 .. batch: 4/5 .. LR: 0.05 .. KL_theta: 0.08 .. Rec_loss: 1515.24 .. NELBO: 1515.32
--------------------------------------------------
Epoch----->98 .. LR: 0.05 .. KL_theta: 0.08 .. Rec_loss: 1515.24 .. NELBO: 1515.32
Eval Doc Completion PPL: 27488.9
Training finished.
Epoch: 99 .. batch: 2/5 .. LR: 0.05 .. KL_theta: 0.07 .. Rec_loss: 1567.92 .. NELBO: 1567.99
Epoch: 99 .. batch: 4/5 .. LR: 0.05 .. KL_theta: 0.07 .. Rec_loss: 1514.45 .. NELBO: 1514.52
--------------------------------------------------
Epoch----->99 .. LR: 0.05 .. KL_theta: 0.07 .. Rec_loss: 1514.45 .. NELBO: 1514.52
Eval Doc Completion PPL: 27819.3
```

Figure 35: Loss values for DistilBERT-based ETM model in the last training epochs.

```
Epoch----->97 .. LR: 0.05 .. KL_theta: 22.4 .. Rec_loss: 1078.73 .. NELBO: 1101.13
Eval Doc Completion PPL: 5763.0
Training finished.
Epoch: 98 .. batch: 2/5 .. LR: 0.05 .. KL_theta: 22.13 .. Rec_loss: 1084.61 .. NELBO: 1106.74
Epoch: 98 .. batch: 4/5 .. LR: 0.05 .. KL_theta: 22.08 .. Rec_loss: 1078.98 .. NELBO: 1101.06
--------------------------------------------------
Epoch----->98 .. LR: 0.05 .. KL_theta: 22.08 .. Rec_loss: 1078.98 .. NELBO: 1101.06
Eval Doc Completion PPL: 5734.6
Training finished.
Epoch: 99 .. batch: 2/5 .. LR: 0.05 .. KL_theta: 21.83 .. Rec_loss: 1068.98 .. NELBO: 1090.81
Epoch: 99 .. batch: 4/5 .. LR: 0.05 .. KL_theta: 22.14 .. Rec_loss: 1078.92 .. NELBO: 1101.06
--------------------------------------------------
Epoch----->99 .. LR: 0.05 .. KL_theta: 22.14 .. Rec_loss: 1078.92 .. NELBO: 1101.06
Eval Doc Completion PPL: 5805.0
Training finished.
```

Figure 36: Loss values for GloVe-based ETM model in the last training epochs.

```
<Training Info>
| Iterations: 1000, Burn-in steps: 100
| Optimization Interval: 10
| Log-likelihood per word: -29.35916
|
<Initial Parameters>
| tw: TermWeight.IDF
| min_cf: 3 (minimum collection frequency of words)
| min_df: 0 (minimum document frequency of words)
| rm_top: 8 (the number of top words to be removed)
| k: 125 (the number of topics between 1 ~ 32767)
| alpha: 0.0008 (hyperparameter of Dirichlet distribution for document-topic)
| eta: 2.82e-05 (hyperparameter of Dirichlet distribution for topic-word)
| seed: 41 (random seed)
| trained in version 0.9.1
|
```

(a)

```
<Parameters>
| alpha (Dirichlet prior on the per-document topic distributions)
|  [0.01890876 0.03327749 0.02756519 0.03001748 0.30038354 0.08986171
|   0.01580108 0.08982162 0.02185369 0.02927051 0.02233893 0.05185439
|   0.01129061 0.40859014 0.02704086 0.0222178  0.01171764 0.01037625
|   0.01144968 0.01047778 0.01313897 0.02922549 0.03404425 0.01335939
|   0.03511122 0.06402308 0.0516989  0.0205155  0.00941426 0.10694552
|   0.01555886 0.08553656 0.01156082 0.01247853 0.01010678 0.00904783
|   0.00786188 0.01544492 0.01146986 0.01177212 0.02530446 0.05188403
|   0.03050941 0.02381381 0.05382099 0.02556203 0.0333138  0.0397423
|   0.01853933 0.01778479 0.0295393  0.01172912 0.00964545 0.17924713
|   0.02588366 0.12758577 0.01703299 0.03511725 0.03091458 0.00981831
|   0.037761   0.09402224 0.02667727 0.00733185 0.02200259 0.01168125
|   0.0204857  0.05925253 0.01524686 0.01839248 0.00926395 0.007622
|   0.01882504 0.01420527 0.03910429 0.07707909 0.02114504 0.02482978
|   0.01362663 0.0744482  0.05517156 0.07428321 0.03127457 0.05984493
|   0.05347418 0.18148741 0.02490239 0.05612065 0.01134892 0.02785396
|   0.01246965 0.04016779 0.01939483 0.03940081 0.02242122 0.00828587
|   0.02271824 0.03552142 0.00846476 0.0135331  0.01023046 0.02416242
|   0.09294128 0.01139615 0.00824781 0.01544748 0.00667433 0.01173942
|   0.0166338  0.04037106 0.04712452 0.00733564 0.02509716 0.01939786
|   0.02895245 0.00797705 0.13582873 0.05636029 0.04633066 0.01138277
|   0.25804842 0.01349785 0.04566184 0.00973929 0.01161844]
| eta (Dirichlet prior on the per-topic word distribution)
|  2.82e-05
```

(b)

Figure 37: Training details for the final Streaming LDA model. Note that these hyper-parameters are shared across all batches.

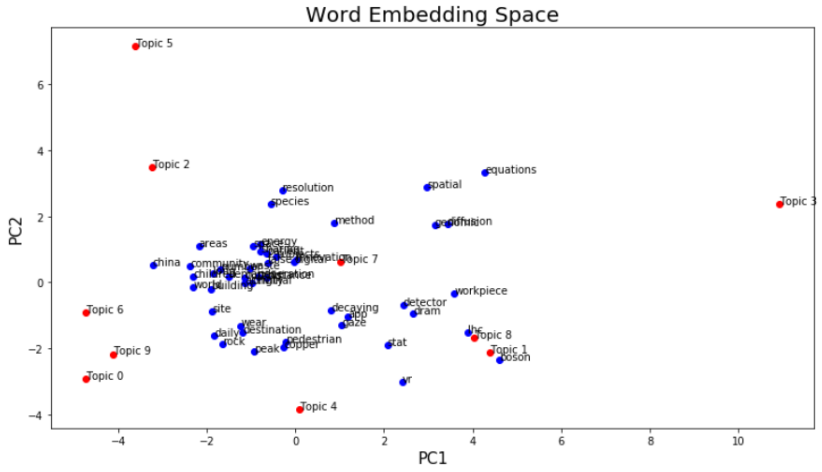


Figure 38: Visualisation of the topic embedding vectors, together with a few randomly selected representative words.

Figure 39: Visualisation of author embedding vectors. The different colors correspond to the respective department of the author.