

**Manuscript version: Author's Accepted Manuscript**

The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

**Persistent WRAP URL:**

<http://wrap.warwick.ac.uk/153608>

**How to cite:**

Please refer to published version for the most recent bibliographic citation information.

**Copyright and reuse:**

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**Publisher's statement:**

Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: [wrap@warwick.ac.uk](mailto:wrap@warwick.ac.uk).

# A Query-Driven Topic Model

Zheng Fang<sup>1</sup>, Yulan He<sup>1,2</sup> and Rob Procter<sup>1,2</sup>

<sup>1</sup>Department of Computer Science, University of Warwick

<sup>2</sup>The Alan Turing Institute, London, UK

{Z.Fang, Yulan.He | Rob.Procter}@warwick.ac.uk

## Abstract

Topic modeling is an unsupervised method for revealing the hidden semantic structure of a corpus. It has been increasingly widely adopted as a tool in the social sciences, including political science, digital humanities and sociological research in general. One desirable property of topic models is to allow users to find topics describing a specific aspect of the corpus. A possible solution is to incorporate domain-specific knowledge into topic modeling, but this requires a specification from domain experts. We propose a novel query-driven topic model that allows users to specify a simple query in words or phrases and return query-related topics, thus avoiding tedious work from domain experts. Our proposed approach is particularly attractive when the user-specified query has a low occurrence in a text corpus, making it difficult for traditional topic models built on word co-occurrence patterns to identify relevant topics. Experimental results demonstrate the effectiveness of our model in comparison with both classical topic models and neural topic models.

## 1 Introduction

Topic modeling aims to infer topics from a collection of documents, where a topic is a salient pattern of the collection and is represented by a distribution over words. The availability in large volume of new sources of unstructured data, such as social media, has presented a challenge to conventional qualitative research methods in the social sciences and humanities and encouraged the exploration of topic modeling as a potential solution (Melville et al., 2019; Hu et al., 2019; Yao and Wang, 2020). In these studies, topic modeling has been applied to questions centered on interpretation and meaning. By analyzing words distribution of topics learnt, researchers can apply inductive reasoning on spe-

cific topics and perform a more in-depth study of related documents, allowing them to identify underlying topical trends and conduct a more thorough analysis of the data.

One limitation of conventional topic modeling approaches in these studies is that they can only learn topics from the whole corpus. However, in some cases, researchers may be interested in topics describing specific concepts or aspects of the corpus. To identify these topics, researchers have to analyze words distribution for all topics, thereby making it very time consuming. Moreover, it could also happen that the target topics may have a very small presence in the data to be detected directly by a topic model. For instance, given a set of posts about health, researchers may wish specifically to analyze the impact of food on health. If the words related to food have a relatively low frequency of occurrence in the posts, then conventional topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) may not find any food-related topics at all. This is caused by the phenomenon of higher order co-occurrence in conventional topic models (Heinrich, 2009), which prevents infrequent words being sampled under the correct topic. While an information retrieval method could be used to find relevant documents, identifying key subtopics discussed in these documents will still be a daunting process.

To handle this limitation, weakly-supervised approaches (Andrzejewski and Zhu, 2009; Nikolenko et al., 2017; Chen et al., 2013; Andrzejewski et al., 2011; Yang et al., 2015) have been proposed as a solution and different types of domain-specific, prior knowledge, such as word correlation (Yang et al., 2015), document and word labels have been introduced. By adding these to the unsupervised topic model, a set of topics describing the domain knowledge can be generated. However, this still requires experts to define the domain knowledge, which may not always be feasible. In addition, the

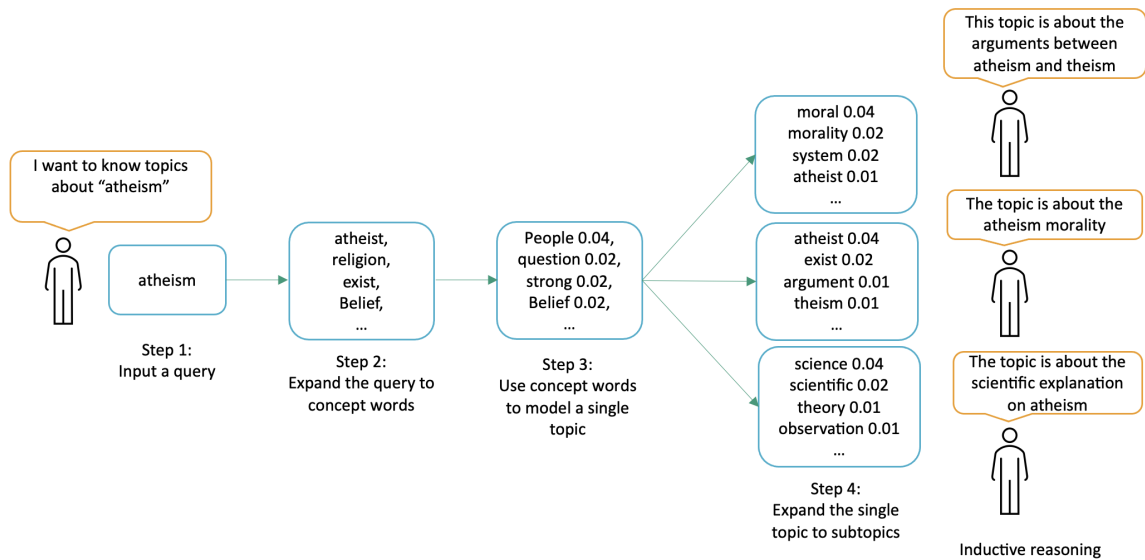


Figure 1: Our proposed model returns topics relevant to a user-input query, in this example, ‘*atheism*’. **Step 1:** user uses a query to define the concept of interest. **Step 2:** a query expansion technique is used to expand the input query to a set of concept words. **Step 3:** the concept words are utilized to generate a single topic. **Step 4:** the single topic is expanded to a set of subtopics. The retrieved concept-topic and subtopic results allow the user to do inductive reasoning and have a more in-depth study of related documents. In Step 3 and Step 4, we present the top weighted words of the topic and their corresponding weights.

aforementioned approaches can only generate one topic relevant to the target concept. It is desirable to distinguish between different contexts about the same concept: for instance, for the concept ‘*Middle East*’, there might be subtopics relating to Middle East conflicts and Middle East resorts, respectively. In our work, we propose a novel approach that automatically generates all subtopics relevant to the target concept.

In our query-driven topic model, a query phrase is used to define the concept of interest. As illustrated in Figure 1, a query expansion technique is first employed to expand the input query to a set of concept words, which are then utilized to first generate a single topic about the concept, and subsequently further expanded to a set of subtopics automatically. In summary, our contributions are four fold: (1) We propose a novel approach which allows users without expertise knowledge to use a short query rather than predefined keywords to detect topics of their interests; (2) Our model is novel in its ability to identify rare topics in text, which would not be possible using existing topic modeling approaches; (3) Our model is built on the Hierarchical Dirichlet Process (HDP) and can therefore automatically infer all subtopics describing the target concept without having to determine the optimal number of topics beforehand; (4) We

evaluate our approach on three datasets and achieve superior performance compared to both traditional hierarchical topic models and neural topic models, both quantitatively and qualitatively.<sup>1</sup>

## 2 Related Work

Earlier work has attempted to solve the problem of identifying specific topics by using prior knowledge. Andrzejewski et al. (2009) expressed domain knowledge with two primitives on word pairs called *Must-Links* and *Cannot-Links*, encoding them using a Dirichlet Forest prior. Topic-in-set knowledge (Andrzejewski and Zhu, 2009) defines ‘*z-labels*’ as prior knowledge and a similar idea was introduced by Nikolenko et al. (2017). First-Order Logic has been proposed as a way to incorporate richer forms of prior knowledge (Andrzejewski et al., 2011). Yang et al. (2015) proposed an efficient method for incorporating domain knowledge and demonstrated significant speed improvement with large datasets. El-Assady et al. (2019) presented a framework that allows users to incorporate the semantics of their domain knowledge in topic models interactively. Gemp et al. (2019) incorporated informative priors in a neural topic model for the purpose of semi-supervised topic modeling. All these approaches

<sup>1</sup>Our source code can be accessed at: <https://github.com/Fitz-like-coding/QDTM>.

require experts to provide domain-specific, prior knowledge, which is problematic for two reasons: different corpora in the same domain may contain different information; and it may be costly to specify all prior knowledge. We take advantage of a query expansion technique and propose an automatic concept words extractor to help user extract prior knowledge.

Our work is also similar to the Hierarchical Topic Model (HTM) (Blei et al., 2004). HTM is a non-parametric topic model that generates topics in a hierarchical structure. In our work, we also propose to generate subtopics from a parent topic. A key difference is that we propose a novel solution to incorporate domain-specific prior knowledge, making it possible to generate desirable topics. This is not the case with HTM. Although attempts were made to introduce prior knowledge in HTM, Perotte et al. (2011) focused on out-of-sample label prediction which is not the focus of our work while Xu et al. (2018) still required experts to define word pairs which is problematic as mentioned earlier.

### 3 Proposed Framework

In outline, our model expands an input query to a set of concept words using a concept words extractor. These concept words are then fed into a two phases framework based on a variant of a Hierarchical Dirichlet Process (HDP) to model all topics relevant to the concept.

#### 3.1 Concept Words Extractor

Given an input query  $q$ , we retrieve a list of documents  $d$  according to the query likelihood score (Ceri et al., 2013),

$$p(d|q) \approx \prod_{i=1}^n p(q_i|d) \quad (1)$$

where  $n$  is the number of tokens in the query and  $p(q_i|d)$  is the probability of query term  $q_i$  in document  $d$ . We define two extraction rules ‘‘AND’’ and ‘‘OR’’ to constrain whether query terms should appear in the same document or not. We then extract concept words from the retrieved documents. We adopt three approaches for our purpose.

**Frequency based extraction (FRE)** The first one simply extracts words with high frequency in the retrieved documents as our concept words:

$$Score(w) = \sum_i TF(w|d_i) \quad (2)$$

where  $n$  is the number of retrieved documents and  $TF(w|d_i)$  is the term frequency of word  $w$  in document  $d_i$ .

**KL-Divergence based extraction (KLD)** The second one is inspired by the query expansion technique (Carpineto et al., 2001). By intuition, words relevant to the input query have a high probability in the retrieved sub-corpus but a low probability in the whole corpus. The score can be defined as:

$$Score(w) = P_R(w) \log \frac{P_R(w)}{P_C(w)} \quad (3)$$

where  $P_R(w)$  is the probability of word  $w$  in the retrieved sub-corpus and  $P_C(w)$  is the probability of word  $w$  in the whole corpus. We extract words with high scores as our concept words.

#### Relevance model with word embedding (REL)

This approach extracts concept words from a word-embedding enhanced relevance model (Diaz et al., 2016). The probability assigned to word  $w$  by the relevance model (Lavrenko and Croft, 2017) is:

$$p(w|RM) = \sum_{d \in R} p(w|d)p(d|q) \quad (4)$$

where  $R$  is the retrieved documents set,  $p(w|d)$  is the probability of word  $w$  in document  $d$  and  $p(d|q)$  is  $d$ 's query likelihood from equation (1). We integrate this model with word embeddings:

$$Score(w) = \lambda p(w|RM) + (1 - \lambda) \text{sim}(w, q) \quad (5)$$

where  $\lambda$  is a hyperparameter and  $\text{sim}(w, q)$  is the normalized similarity between word  $w$  and the input query  $q$ . For each term in the vocabulary list, we calculate its similarity with the input query. We then take the top  $k$  most similar terms and normalize their similarity values. If  $w$  is among the top  $k$  similar terms,  $\text{sim}(w, q)$  would get the normalized similarity value. Otherwise,  $\text{sim}(w, q) = 0$ .

#### 3.2 Query-Driven Topic Model

We propose a two-phase framework based on HDP, which is a nonparametric Bayesian model that can automatically infer the number of topics in a corpus (Teh et al., 2005). It assumes a restaurant (i.e., a document) has a set of tables and serves dishes (i.e., topics) from a global menu. A single dish is only served at a single table for all customers (i.e., words) who sit at that table.

In the first phase, the model infers one topic for each concept, along with other irrelevant topics.

We define this topic as the “*parent topic*” in later sections. We denote this parent topic of a concept corresponding to the input query  $q$  as  $\tilde{z}_q$ . We incorporate prior knowledge into HDP by fixing the topic index for concept words in all documents. For words from concept words  $W_q$  corresponding to the input query  $q$ , the topic index  $z$  are known and remain fixed as  $\tilde{z}_q$ , and the probability for sampling an existing table  $t$  for a word  $w_{ji}$  at document  $j$  and position  $i$  in the Gibbs sampling process is:

$$p(t_{ji} = t \mid t^{-ji}, k) \propto \mathbb{1}_{\mathbb{1}}(w_{ji}, k_{jt}) n_{jt}^{-ji} f_{k_{jt}}^{-w_{ji}}(w_{ji}) \quad (6)$$

where  $k_{jt}$  is the topic assignment of table  $t$  at document  $j$  and  $f_{k_{jt}}^{-w_{ji}}(w_{ji})$  is the probability of  $w_{ji}$  assigned to topic  $k_{jt}$  after removing the current word and  $\mathbb{1}_{\mathbb{1}}(w_{ji}, k_{jt})$  is an indicator function, which takes on value 0 if  $w_{ji} \in W_q$  and  $k_{jt} \neq \tilde{z}_q$  and 1 otherwise.  $n_{jt}^{-ji}$  denotes the number of words in document  $j$  at table  $t$  except the current word. The probability for sampling a new table  $t^{new}$  is:

$$p(t_{ji} = t^{new} \mid t^{-ji}, k) \propto \alpha p(w_{ji} \mid t^{-ji}, t^{new}, \mathbf{k}) \quad (7)$$

where

$$p(w_{ji} \mid t^{-ji}, t^{new}, \mathbf{k}) = \sum_{k=1}^K \frac{m_k}{m. + \gamma} f_k^{-w_{ji}}(w_{ji}) + \frac{\gamma}{m. + \gamma} f_{k^{new}}^{-w_{ji}}(w_{ji}) \quad (8)$$

Here,  $m_k$  denotes the number of tables of topic  $k$  and  $m.$  denotes the total number of tables.  $\gamma$  and  $\alpha$  are the hyperparameters of the model.  $f_{k^{new}}^{-w_{ji}}(w_{ji}) = \frac{1}{|V|}$  is the prior density of  $w_{ji}$  where  $|V|$  is the vocabulary size of the dataset. If the sampled table is a new table, we sample an existing topic  $k_{jt^{new}}$  from:

$$p(k_{jt^{new}} \mid t, k^{-jt^{new}}) \propto \mathbb{1}_{\mathbb{1}}(w_{ji}, k_{jt}) m_k f_k^{-w_{ji}}(w_{ji}) \quad (9)$$

and probability for sampling a new topic  $k^{new}$  is:

$$p(k_{jt^{new}} = k^{new} \mid t, \mathbf{k}^{-jt^{new}}) \propto \gamma f_{k^{new}}^{-w_{ji}}(w_{ji}) \quad (10)$$

In the second phase, the model expands the parent topic of each concept produced in the first phase to a set of subtopics. Let  $W_{\tilde{z}_q}$  be the words assigned to the parent topic  $\tilde{z}_q$  in the first phase, the probability for sampling an existing table  $t$  for a word  $w_{ji}$  in the Gibbs sampling process is:

$$p(t_{ji} = t \mid t^{-ji}, k) \propto \mathbb{1}_{\mathbb{2}}(w_{ji}, k_{jt}) n_{jt}^{-ji} f_{k_{jt}}^{-w_{ji}}(w_{ji}) \quad (11)$$

where  $\mathbb{1}_{\mathbb{2}}(w_{ji}, k_{jt})$  is an indicator function that takes on value 1 if  $w_{ji} \in W_{\tilde{z}_q}$  and  $k_{jt} = \tilde{z}_q$  and

0 otherwise. Probability for sampling a new table  $t^{new}$  is:

$$p(t_{ji} = t^{new} \mid t^{-ji}, k) \propto \mathbb{1}_{\mathbb{2}}(w_{ji}, k_{jt}) \alpha p(w_{ji} \mid t^{-ji}, t^{new}, \mathbf{k}) \quad (12)$$

where

$$p(w_{ji} \mid t^{-ji}, t^{new}, \mathbf{k}) = \sum_{k=1}^K \frac{m_k}{m. + \gamma} f_k^{-w_{ji}}(w_{ji}) + \frac{\gamma}{m. + \gamma} \hat{f}_{k^{new}}^{-w_{ji}}(w_{ji}) \quad (13)$$

and  $\hat{f}_{k^{new}}^{-w_{ji}}(w_{ji}) = \frac{1}{|W_{\tilde{z}_q}|}$  is the prior density of  $W_{\tilde{z}_q}$  where  $|W_{\tilde{z}_q}|$  is the vocabulary size of  $W_{\tilde{z}_q}$ . For a new table, probability for sampling an existing topic  $k$  is:

$$p(k_{jt^{new}} = k \mid t, \mathbf{k}^{-jt^{new}}) \propto \mathbb{1}_{\mathbb{2}}(w_{ji}, k_{jt}) m_k f_k^{-w_{ji}}(w_{ji}) \quad (14)$$

and probability for sampling a new topic  $k^{new}$  subordinate to the parent topic  $\tilde{z}_q$  is:

$$p(k_{jt^{new}} = k^{new} \mid t, \mathbf{k}^{-jt^{new}}) \propto \gamma \hat{f}_{k^{new}}^{-w_{ji}}(w_{ji}) \quad (15)$$

The model automatically decides the number of subtopics and we treat the subtopics produced as the final topics relevant to the target concept.

### Incorporating Generalized Pólya Urn scheme

To make topics more interpretable, we incorporate word-embeddings by the Generalized Pólya Urn scheme (Li et al., 2016). Pólya Urn scheme is introduced for colored balls and urns. In the Generalized Pólya Urn scheme, when we draw a ball of a particular color, two balls of the same color are put back along with a certain number of balls of the similar colors. In topic modeling context, a topic can be viewed as an urn while a word can be viewed as a ball in a certain color and its semantically related words can be viewed as balls of similar colors. Every time we sample a word  $w$  under a parent topic  $\tilde{z}_q$ , we increase the probability of sampling  $w$  under  $\tilde{z}_q$ , as well as its semantically related concept words. Given pre-trained word embeddings, we calculate the cosine similarity between word  $w_i$  and concept word  $w_q \in W_q$ . We then construct a word semantic relatedness matrix  $\mathcal{M}$  (Li et al., 2016), consisting of all word pairs whose cosine similarity is greater than a predefined threshold. We then construct a promotion matrix  $A$

whose elements are defined as:

$$A_{i,q} = \begin{cases} 1, & \text{if } (w_i, w_q) \in \mathcal{M} \text{ and } w_i = w_q \\ u, & \text{if } (w_i, w_q) \in \mathcal{M} \text{ and } w_i \neq w_q \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

where  $u \in (0, 1)$  is a predefined promotion weight. When we sample a word  $w$  under topic  $\tilde{z}_q$ , we also promote all its semantically related concept words based on the amount of promotion in  $A$ .

**Word filtering** Inspired by Wang et al. (2020), we propose a word filtering strategy. Word filtering can be used to prevent words that have weak ties with the sampled topic being promoted. For a word  $w$  at  $i^{\text{th}}$  Gibbs sampling iteration, its semantic cohesion to topic  $k$  is:

$$CV[k, w_i] = \sum_{m=1}^M p^i(k, m) \cdot \cos(w_i, RW^i(k, m)) \quad (17)$$

where  $p^i(k, m)$  is the probability of  $m^{\text{th}}$  representative word in topic  $k$  at  $i^{\text{th}}$  iteration and  $M$  is the number of representative words predefined. The representative words of topic  $k \neq \tilde{z}_q$  at  $i^{\text{th}}$  Gibbs sampling iteration are defined by the words ranked by the topic-word probability in the descending order.  $\cos(w_i, RW^i(k, m))$  is the cosine similarity between word  $w_i$  and the  $m^{\text{th}}$  representative word of topic  $k$  at  $i^{\text{th}}$  iteration. The representative words of  $\tilde{z}_q$  are simply its concept words.

For the semantic cohesion of word  $w$  with different topics  $CV[\cdot, w]$ , we map  $CV[\cdot, w]$  into an arithmetic progression  $\tilde{C}V[\cdot, w]$  ranging from 0 to 1.0 (Wang et al., 2020). We use the following equation to decide if the GPU is applied to  $w$ :

$$S_{j,w} \sim \text{Bernoulli}(\lambda_{w,k_w}) \quad (18)$$

$$\lambda_{w,k} = \frac{\tilde{C}V[k, w]}{\tilde{C}V_{\max}[k, w]}$$

where  $S_{j,w}$  indicates whether GPU is applied to word  $w$  given document  $j$  and  $\tilde{C}V_{\max}[k, w]$  is the maximal semantic cohesion among all topics.

We present the details of the Gibbs sampling process of the first phase of our model in Algorithm 1. We omit the details of the second phase of our model since it is similar to the first phase. The details of the functions Initialize( $\cdot$ ) and UpdateCounter( $\cdot$ ) can be found in Appendix C.

---

### Algorithm 1: Query-driven topic model.

---

**Input:** initial topic number  $K$ , hyperparameters  $\alpha, \beta, \gamma$ , word semantic relatedness matrix  $\mathcal{M}$ , documents  $D$ , and Concept words  $W_Q$ ,  
**Output:** The posterior topic-word distribution  
Initialize( $K, D, W_Q$ );  
/\* first phase \*/;  
**foreach** iteration **do**  
    Update word-topic coherence using Eq.17  
    **foreach** document  $j \in D$  **do**  
        **foreach** position  $i \in j$  **do**  
            Assign table  $t \leftarrow t_{ji}$   
            Assign topic  $k \leftarrow k_{jt}$   
            UpdateCounter( $S_{j,w_{ji}}, t, k, False$ );  
             $t_{ji} \leftarrow t \sim p(t_{ji} = t \mid t^{-ji}, k)$   
            (Eq.6-8)  
            **if**  $t == t^{\text{new}}$  **then**  
                 $k_{jt} \leftarrow k \sim p(k_{jt}^{\text{new}} = k \mid$   
                 $t, k^{-jt^{\text{new}}})$  (Eq.9-10)  
            **end**  
             $S_{j,w_{ji}} \leftarrow \text{updateGPUFlag}(j, w_{ji})$   
            based on Eq.18  
            UpdateCounter( $S_{j,w_{ji}}, t_{ji}, k_{jt}, True$ )  
        **end**  
    **end**  
**end**

---

## 4 Experiments

We conducted our experiments by two steps. In the first step, we evaluated the quality of the parent topics from the first phase of our model. In the second step, we evaluated the quality of the subtopics from the second phase of our model.

### 4.1 Setup

**Datasets** We conducted our experiments on three datasets: 20Newsgroup<sup>2</sup> contains around 18k newsgroup posts on 20 topics; TagMyNews<sup>3</sup> contains around 32k short English news from 7 categories; SearchSnippets (Xu et al., 2017) contains 12k short web search snippets from 8 categories.

**Baselines** We compared our model with six baselines: LDA (Blei et al., 2003) is a widely used topic model; DF-LDA (Andrzejewski et al., 2009) incorporates domain knowledge in LDA with Must-Links and Cannot-Links; SCLDA (Yang et al., 2015) expresses prior knowledge as sparse constraints; ISLDA (Nikolenko et al., 2017) fixes topic index  $z$  for certain keywords in all documents. AVITM (Srivastava and Sutton, 2017) is a neural topic model based on autoencoding Variational Inference. We also compared our model with BERT (Devlin et al., 2018), a well-known neural language

<sup>2</sup><http://qwone.com/~jason/20Newsgroups/>

<sup>3</sup><http://acube.di.unipi.it/tmn-dataset/>

model, to test its document retrieval ability. To evaluate the quality of our subtopics towards the target concepts, we compared our model with HTM (Blei et al., 2004), which is also a nonparametric Bayesian model that can generate subtopics from higher level topics.

**Parameterization** We set  $\alpha = 0.5$ ,  $\beta = 0.1$  for DF-LDA and  $\alpha = 0.1$ ,  $\beta = 0.01$  for SCLDA as suggested by the original papers. We set  $\alpha = 1/K$ ,  $\beta = 1/K$  for LDA and ISLDA, where  $K$  is the number of topics pre-set for the models, and found it outperforms the original settings. We set  $\alpha = 0.1$ ,  $\gamma = 0.1$  and  $\eta = 0.01$  for HTM as suggested by the original paper and set the topic hierarchy depth to 3, to make it easier to compare with our model since topics from the second level of HTM can be considered as the parent topics and those from the third level as the subtopics of the parent topics. We set  $\alpha = 1.0$ ,  $\beta = 0.5$  and  $\gamma = 1.5$  as in the original HDP paper for our query-driven model, and set the threshold for the cosine similarity used for the Generalized Pólya Urn scheme to 0.5 and the promotion weight  $u$  to 0.3. The number of representative words  $M$  for the word filtering strategy was set to 10.  $\lambda$  for the REL query expansion technique was set to 0.5 and  $k$  was set to 100. In our experiments, we treated each category as a concept and determined the number of topics for each baseline model based on the number of categories in the datasets. For example, if a dataset had 16 categories, we set the number of topics to 17, using an extra one representing irrelevant information.

For all baseline models, we asked an expert to provide prior knowledge. Each category in a dataset was associated with 10 keywords provided by the expert. For DF-LDA, we converted keywords to must-links. Since LDA, DF-LDA, AVITM and HTM cannot reveal the relationship between a concept and the generated topics directly, we need a further step to find the relationship between them. We calculated the average pairwise cosine similarity of the keywords and the top-10 word embeddings of each topic, and chose the topic with the highest similarity as the target topic of the concept. For HTM, we use the topics from the second level of its generated topic hierarchy. For our model, we used query phrases to represent the main concept of each category. Query phrases were interpreted directly from category names, e.g., we used “*computer graphics*” to represent the category “`comp.graphics`” in the 20News group

dataset. We removed categories that do not have meaningful names due to the difficulty of defining the query phrases for these categories, e.g., “`talk.politics.misc`” in the 20News group dataset. We then selected the top 10 concept words of each query based on the scores from the concept words extractor. We list expert-defined keywords and query phrase for each category in Appendix A. All models were trained until convergence. For BERT, we simply used the query phrases to retrieve relevant documents. We ran each model five times and present their average performance.

## 4.2 Parent Topic Evaluation

We evaluated the quality of parent topics of our model in terms of document classification, topic coherence and document retrieval performance. For document classification, a logistic regression classifier with default parameter settings was used. We used the topic distribution of each document as the input and conducted five-fold cross-validation. The topic distribution of a document represents the probability of each topic in a document. The quality of the topics can be assessed by the accuracy of text classification using the topic-level representation. A better classification accuracy means better latent semantic representations of the topics, indicating the learnt topics are more discriminative and representative. For topic coherence measure, we followed Roder et al. (2015) and used the best performing topic coherence measure C\_V based on the external corpus (Wikipedia). We focused on the top 10 words of our parent topics and used the Palmetto library algorithm (Röder et al., 2015). Higher coherence indicates better topic interpretability. For document retrieval, we adopted the metric “precision@K” (P@K), which corresponds to the number of relevant results among the top  $K$  documents. We retrieved documents of each topic based on the probability of the topic in the documents  $p(z|d)$ . If a topic can describe the target concept well, then the top retrieved documents should be relevant to the concept. In our experiments, we know the ground-truth number of documents belonging to each category, therefore we set  $K$  for each concept to the actual number of documents from the corresponding category. We only considered the parent topics and reported the average results. A higher score indicates the model retrieves more concept-relevant documents, which is important when a researcher wants to do a more

Model	20news			TagMyNews			SearchSnippets		
	Acc	coherence	Precision@K	Acc	coherence	Precision@K	Acc	coherence	Precision@K
LDA	0.650	0.420	0.588	0.781	0.384	0.687	0.804	0.390	0.696
DFLDA	0.623	0.421	0.562	0.772	0.386	0.649	0.795	0.390	0.644
SCLDA	0.666	0.402	0.622	0.804	0.418	0.745	0.816	0.414	0.796
ISLDA	0.680	0.406	0.645	0.801	0.411	0.729	0.845	0.421	0.804
BERT	–	–	0.156	–	–	0.300	–	–	0.261
AVITM	0.504	<b>0.494</b>	0.381	0.728	0.482	0.583	0.678	0.461	0.596
Query-driven model(FRE)	<b>0.707</b>	0.433	<b>0.679</b>	0.807	0.429	0.716	0.858	0.469	0.807
Query-driven model(REL)	0.601	0.452	0.557	<b>0.837</b>	0.444	0.749	0.851	0.501	0.780
Query-driven model(KLD)	0.705	0.435	0.677	<b>0.828</b>	0.414	<b>0.755</b>	0.860	0.465	0.811
+ expert keywords	0.690	0.430	0.659	0.817	<b>0.493</b>	0.747	<b>0.864</b>	<b>0.503</b>	<b>0.841</b>
– word filtering	0.706	0.434	0.678	0.823	0.413	0.744	0.859	0.432	0.811
– GPU	0.698	0.400	0.671	0.818	0.389	0.730	0.849	0.413	0.808

Table 1: Parent topic evaluation results for three datasets. FRE indicates frequency based query expansion; REL indicates Word embedding enhanced query expansion; KLD indicates KL-Divergence based query expansion. BERT here is only for document retrieval purpose therefore we can’t present the accuracy and coherence in the table.

in-depth study of related documents.

Table 1 shows the performance of our models using three different query expansion techniques as well as only using expert-defined key words as prior knowledge. It can be observed that our models using the FRE and KLD query expansion techniques outperform all baselines except AVITM on almost all measures, though our model using the FRE query expansion technique has slightly worse document retrieval performance on the TagMyNews dataset. Although the model using REL is not as competitive as the models using FRE and KLD on the 20newsdataset, it has the highest coherence scores on all the datasets, despite not using any expert-defined keywords as prior knowledge. This shows that combining word embeddings in query expansion can help produce more coherent prior knowledge. Although AVITM has better coherence score than our model on the 20newsdataset and TagMyNews datasets, its poor document retrieval performance indicates it is unable to find documents relevant to the target concept. Comparing our query-driven model with or without using expert-defined keywords, it achieves better coherence scores on the 20Newsdataset without expert-defined keywords, though it performs slightly worse on the other two datasets.

As for document classification and document retrieval, our KLD-based model has better performance than expert-defined keywords on 20newsdataset and TagMyNews, but does not work well on SearchSnippets. This may be because our concept words extractor does not work well on short texts. The concept words extracted from TagMyNews

and SearchSnippets are not as competitive as expert defined keywords. In addition, different interpretations of the same concept word may also compromise the performance. Interestingly, we also observed that BERT does not work well on these datasets. Possibly this is because we are using short query phrases to represent the concepts and BERT only works well for long queries. A short query may not give enough information about the concept that’s why we adopted query expansion and topic modeling approaches. Table 2 shows example concept-specific topics extracted from the TagMyNews dataset. It can be seen that extracted topics are closely related to their respective concept phrase. Topic extraction results on the other two datasets are shown in Appendix B.

Concept Phrase	Top 10 words
<b>business</b>	profit, business, bank, sell, usa, sale, credit, price, billion, stock
<b>entertainment</b>	week, show, star, theater, pop, time, film, tv, sony, wedding
<b>health</b>	disease, health, study, care, risk, drug, cancer, insurance, people, usa
<b>technology</b>	google, apple, technology, ipad, china, company, online, intel, service, network
<b>sport game</b>	game, play, playoff, final, boston, win, series, scored, season, sport

Table 2: Target concept topics for TagMyNews dataset.

**Ablation study:** We also studied the effectiveness of two major components in the proposed model: 1) GPU to incorporate word embeddings;



Model	20news			TagMyNews			SearchSnippets		
	Diversity	Cohesion	overall	Diversity	Cohesion	overall	Diversity	Cohesion	overall
HTM	<b>0.94</b>	0.54	0.51	<b>0.93</b>	0.53	0.49	<b>0.86</b>	0.49	0.42
Query-driven model	0.71	<b>0.79</b>	<b>0.56</b>	0.68	<b>0.79</b>	<b>0.54</b>	0.74	<b>0.76</b>	<b>0.56</b>

Table 3: Topic diversity and cohesion results for subtopics.

2) word filtering to remove unimportant words. The last two rows in Table 1 show the performance of our model using KLD query expansion technique without GPU and word filtering components. These show that GPU has a big impact for coherence and can help improve other measures in some extents, while removing word filtering reduces performance on all measures.

### 4.3 Subtopics Evaluation

We used our model with the KLD-query expansion technique in this evaluation. We dropped subtopics that have prevalence of less than 0.5% in the corpus as these subtopics usually are not of interest. We evaluated the quality of subtopics in terms of topic diversity and topic cohesion<sup>4</sup>. Topic diversity measures how much a subtopic overlaps with each other. We define it to be the percentage of the unique words in the top 25 words of all subtopics subordinate to the same parent topic (Dieng et al., 2020). Higher diversity indicates more varied topics, while lower diversity indicates more redundant topics. Topic cohesion measures the relevance between the subtopics and the parent topic. We define it to be the cosine similarity between the parent topic embedding and the subtopic embedding. We can get the topic embedding as the weighted summing of the embeddings of its top 10 associated word. We combine these two metrics and define the overall quality of a subtopic as their product.

We report the results in Table 3. It shows that our model outperforms HTM by topic cohesion measures on all datasets, though with lower topic diversity scores. The high cohesion score indicates that the subtopics of our model is highly relevant to the target concept. By taking both measures into a account, our model achieves relatively better performance. It is expected since we incorporate domain prior knowledge into our model.

<sup>4</sup>Note that *topic cohesion* is different from *topic coherence* as *topic cohesion* measure the relevant between a subtopic and its parent topic.

### 4.4 Qualitative Evaluation

We present the qualitative evaluation results in this section. We show a set of topics produced by our model from the 20newsgroup dataset in Table 4. The input concept phrases are shown in the left side of the table. For the concept phrase “*atheism*”, which means the absence of belief in the existence of deities, we can see that our parent topic is highly relevant to it. The topic words like “*question*”, “*belief*”, “*god*” and “*lack*” clearly indicate that the topic is related to the arguments about God. By looking at the subtopics, we can easily see the first subtopic is about the atheism morality, the second subtopic is about the arguments between atheism and theism, and the third subtopic is about the scientific explanation on atheism. For the concept phrase “*for sale*”, which means selling an item in a cheaper price, our parent topic includes many relevant words, such as “*sell*”, “*sale*”, “*sold*” and “*price*”. The inclusion of “*box*” is less easy to explain, but could be related to product packaging. As expected, our subtopic reveals a sub-aspect about the concept that can not be identified directly from the parent topic: the email subscription for the latest news. This is reasonable, since merchants usually use email to provide customers with information about the latest products. The words like “*interested*”, “*mail*”, “*send*”, “*information*” and “*original*” provide more information about the concept.

The last row of Table 4 presents the topics of the low occurrence query “*business*”, appeared only 294 times in the corpus, which is extremely low compared with the majority of other words in the corpus. LDA and HTM are unable to generate relevant topics due to the aforementioned “higher order co-occurrence” issue, but our model can produce reasonable topics. The topic words “*encryption*”, “*key*”, “*phone*”, “*company*” and “*business*” in the parent topic shows that the topic is related to data encryption for business. By looking at the subtopics of the topic, we can get a rough idea that the first subtopic is about a Japanese phone company, since the top weighted words include

Concept phrase	Parent topic	Subtopics
<b>atheism</b>	people, question, strong, belief, make, god, thing, point, religion, lack	moral, system, morality, society, nature, objective, human, dream, animal, action people, question, thing, make, point, god, argument, belief, claim, true science, scientific, theory, result, observation, scientist, experiment, hypothesis, methodology, model
<b>for sale</b>	sell, sale, original, sold, interested, included, price, offer, box, cd	sale, offer, price, sell, original, shipping, box, condition, interested, cd list, interested, send, mail, address, post, email, original, information, call
<b>business</b>	3do, government, key, phone, technology, company, business, chip, encryption, clipper	company, phone, 3do, technology, business, number, system, japanese, computer, make government, clipper, encryption, chip, system, nsa, phone, people, security, key key, chip, algorithm, number, clipper, encryption, de, escrow, system, secret

Table 4: Parent topic and the subtopics of the concepts “*atheism*”, “*for sale*” and “*business*” for the 20newsgroup dataset.

“*japanese*”, “*company*”, “*phone*”, “*technology*”. This makes sense since phone companies usually have a strong requirement for encryption. The second and third subtopics are about encryption algorithms, since the top weighted words include “*chip*”, “*nsa*”, “*key*” and “*algorithm*”. We further verified that our interpretation is correct by looking at the top weighted documents of the topics. This confirms that our model has potential for use in real world applications.

## 5 Conclusions and Future Work

We presented a novel, query-driven topic model to help identify topics of interest in large datasets. Instead of asking experts to define keywords for these topics, we implemented a concept words extractor to automatically extract concept words and used the GPU model, incorporating word-filtering, to improve interpretability and performance. To distinguish between different contexts for the same concept, we further introduced a subtopic modeling procedure. The procedure can automatically infer all subtopics without having to determine the optimal number of subtopics beforehand. Experimental

results on three benchmark datasets demonstrate the model’s promise. In the future, we plan to evaluate our model’s performance using real-world, qualitative analysis use cases.

## Acknowledgments

This work is partly funded by the EPSRC (grant no. EP/T017112/1, EP/V048597/1). ZF receives the PhD studentship jointly funded by the University of Warwick and China Scholarship Council. YH and RP are supported by Turing AI Fellowships funded by UK Research and Innovation (UKRI) (grant nos. EP/V020579/1 and EP/N510129/1 respectively).

## References

- David Andrzejewski and Xiaojin Zhu. 2009. Latent dirichlet allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 43–48.
- David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th annual international conference on machine learning*, pages 25–32.
- David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. 2011. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *Twenty-Second International Joint Conference on Artificial Intelligence*.
- David M Blei, Thomas L Griffiths, Michael I Jordan, and Joshua B Tenenbaum. 2004. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16(16):17–24.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Claudio Carpineto, Renato De Mori, Giovanni Romano, and Brigitte Bigi. 2001. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)*, 19(1):1–27.
- Stefano Ceri, Alessandro Bozzon, Marco Brambilla, Emanuele Della Valle, Piero Fraternali, and Silvia Quarteroni. 2013. An introduction to information retrieval. In *Web information retrieval*, pages 3–11. Springer.
- Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Leveraging multi-domain prior knowledge in topic models. In *Twenty-Third International Joint Conference on Artificial Intelligence*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*.
- Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.
- Mennatallah El-Assady, Rebecca Kehlbeck, Christopher Collins, Daniel Keim, and Oliver Deussen. 2019. Semantic concept spaces: Guided topic model refinement using word-embedding projections. *IEEE transactions on visualization and computer graphics*, 26(1):1001–1011.
- Ian Gemp, Ramesh Nallapati, Ran Ding, Feng Nan, and Bing Xiang. 2019. Weakly semi-supervised neural topic models.
- Gregor Heinrich. 2009. A generic approach to topic models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 517–532. Springer.
- Nan Hu, Ting Zhang, Baojun Gao, and Indranil Bose. 2019. What do hotel customers complain about? text analysis using structural topic model. *Tourism Management*, 72:417–426.
- Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM New York, NY, USA.
- Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 165–174.
- Sophie Melville, Kathryn Eccles, and Taha Yasseri. 2019. Topic modeling of everyday sexism project entries. *Frontiers in Digital Humanities*, 5:28.
- Sergey I Nikolenko, Sergei Koltcov, and Olessia Koltsova. 2017. Topic modelling for qualitative studies. *Journal of Information Science*, 43(1):88–102.
- Adler Perotte, Frank Wood, Noemie Elhadad, and Nicholas Bartlett. 2011. Hierarchically supervised latent dirichlet allocation. *Advances in neural information processing systems*, 24:2609–2617.
- Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.
- Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.
- Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2005. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.
- Rui Wang, Deyu Zhou, and Yulan He. 2020. Optimising topic coherence with weighted  $\text{po}^{\prime}$ lya urn scheme. *Neurocomputing*, 385:329–339.
- Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31.

- Yueshen Xu, Jianwei Yin, Jianbin Huang, and Yuyu Yin. 2018. Hierarchical topic modeling with automatic knowledge mining. *Expert Systems with Applications*, 103:106–117.
- Yi Yang, Doug Downey, and Jordan Boyd-Graber. 2015. Efficient methods for incorporating knowledge into topic models. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 308–317.
- Fang Yao and Yan Wang. 2020. Tracking urban geotopics based on dynamic topic model. *Computers, Environment and Urban Systems*, 79:101419.

## Appendix A: Query and keywords for each category

Concept Phrase	Expert Defined Keywords									
<b>atheism</b>	Agnosticism	theism	deism	islam	paganism	moral	atheist	religions	argument	exist
<b>computer graphics</b>	image	digital	visual	3d	2d	visualization	print	geometry	synthesizing	processing
<b>pc hardware</b>	cpu	monitor	keyboard	memory	card	sound	speakers	motherboard	power	pc
<b>mac hardware</b>	touchpad	touchbar	drive	apple	mac	ram	gpu	system	sensors	physical
<b>for sale</b>	product	mail	discount	bargain	shopping	price	sale	propertise	rent	summer
<b>automobile</b>	car	vehicle	transportation	wheel	tire	road	parking	gasoline	energy	driver
<b>motorcycles</b>	bike	scooters	mopads	motorbikes	trowel	commute	helmet	ride	speed	harley
<b>baseball</b>	player	ball	small	hit	team	fielding	batting	runs	nbl	baseball
<b>hockey</b>	puck	nhl	hockey	ice	rink	canada	rubber	curve	skater	guard
<b>encrypt</b>	encoding	decryption	cryptographic	secure	plaintext	ciphertext	key	algorithm	pseudo	private
<b>electronics</b>	equipment	science	electricity	wire	console	computer	outlet	engineering	power	voltage
<b>medicine</b>	medicine	surgery	hospital	climic	doctor	nurse	healthcare	symtoms	prescription	pharmacy
<b>space</b>	rocket	nasa	astronomy	explore	moon	outerspace	spaceship	telescope	satellite	orbit
<b>christian</b>	belief	faith	church	christianity	ethics	culture	ritual	Jesus	bible	truth
<b>guns</b>	law	regulation	usa	victim	murder	violence	litigation	debate	firearms	legal
<b>middle east</b>	israel	Iran	Iraq	war	territory	turkey	attack	soldier	turkey	government

Table A1: Concept phrases and expert defined keywords for the 20NewsGroup dataset.

Concept Phrase	Expert Defined Keywords									
<b>business</b>	bank	stock	market	business	economy	financial	investor	profit	price	deal
<b>entertainment</b>	film	movie	music	tv	theater	festival	actor	show	book	hollywood
<b>health</b>	drug	health	cancer	patient	disease	medical	hospital	healthcare	doctor	treatment
<b>technology</b>	apple	google	sony	facebook	internet	mobile	ipad	technology	microsoft	phone
<b>sport game</b>	league	win	player	team	tournament	game	playoff	sport	championship	point

Table A2: Concept phrases and expert defined keywords for the TagMyNews dataset.

Concept Phrase	Expert Defined Keywords									
<b>Business</b>	bank	stock	market	business	economy	financial	investor	profit	price	deal
<b>Computers</b>	computer	software	programming	parallel	computing	memory	hardware	driver	cpu	processor
<b>Culture Arts Entertainment</b>	movie	music	art	film	artist	museum	fashion	culture	imdb	actor
<b>Education Science</b>	research	science	journal	university	student	education	scientific	mathematics	theory	school
<b>Car Engineering</b>	engine	electrical	car	wheel	model	automobile	industrial	vehicle	cylinder	jet
<b>Health</b>	drug	health	cancer	patient	disease	medical	hospital	healthcare	doctor	treatment
<b>Politics Society</b>	political	party	democracy	government	republic	parliamentary	representative	president	communist	congress
<b>Sports</b>	league	football	player	team	tournament	game	basketball	sport	hockey	championship

Table A3: Concept phrases and expert defined keywords for the SearchSnippets dataset.

## Appendix B: Topics generated for target concepts

Concept Phrase	Top 10 words									
<b>atheism</b>	people	question	thing	god	strong	point	make	belief	argument	evidence
<b>computer graphics</b>	support	file	version	image	list	graphic	program	information	screen	address
<b>pc hardware</b>	drive	pc	disk	scsi	software	modem	port	hard	controller	system
<b>mac hardware</b>	card	mac	monitor	apple	system	video	problem	write	chip	work
<b>for sale</b>	sell	sale	sold	original	interested	included	price	offer	box	cd
<b>automobile</b>	car	auto	automobile	engine	ford	problem	mile	v6	oil	dealer
<b>motorcycles</b>	bike	riding	ride	motorcycle	rider	battery	buying	dog	back	dod
<b>baseball</b>	run	baseball	game	pitcher	year	hit	player	average	team	good
<b>hockey</b>	game	hockey	team	nhl	night	goal	player	coach	cup	year
<b>encrypt</b>	key	message	chip	encryption	government	clipper	algorithm	system	phone	encrypted
<b>electronics</b>	company	power	line	led	electronics	electronic	circuit	output	work	signal
<b>medicine</b>	patient	medical	treatment	doctor	disease	study	clinical	medicine	food	effect
<b>space</b>	space	nasa	shuttle	launch	satellite	station	moon	1st	cost	orbit
<b>christian</b>	god	christ	church	christian	love	word	bible	jesus	protestant	truth
<b>guns</b>	gun	control	weapon	child	people	police	fire	law	handgun	amendment
<b>middle east</b>	israel	arab	armenian	jew	israeli	muslim	people	middle	east	war

Table B1: Topics of target concepts for the 20NewsGroup dataset.

Concept Phrase	Top 10 words									
<b>Business</b>	business	management	marketing	trade	service	market	law	export	job	stock
<b>Computers</b>	computer	computing	software	web	application	system	programming	apple	memory	chip
<b>Culture Arts Entertainment</b>	art	culture	music	american	artist	tradition	history	ancient	museum	band
<b>Education Science</b>	science	education	research	scientific	undergraduate	biology	journal	school	university	fiction
<b>Car Engineering</b>	car	model	engineering	automobile	engine	wheel	auto	electrical	product	motor
<b>Health</b>	health	disease	care	cancer	public	nutrition	information	medical	gov	drug
<b>Politics Society</b>	political	party	democracy	system	military	politics	government	conflict	gov	war
<b>Sports</b>	football	team	soccer	game	sport	hockey	news	tennis	score	player

Table B2: Topics of target concepts for the SearchSnippets dataset.

## Appendix C: Algorithms

---

### Algorithm 2: Initialize( $K, D, W_Q$ )

---

```

foreach document  $j \in D$  do
   $k \leftarrow z \neq \tilde{z}_Q \sim \text{Multinomial}(1/K)$ ;
  foreach position  $i \in j$  do
    foreach  $W_q \in W_Q$  do
      if  $w_{ji} \in W_q$  then
         $k \leftarrow \tilde{z}_q$ ;
        break;
      end
    end
  end
   $t \leftarrow 0$ ;
  foreach position  $i \in j$  do
     $S_{j,w_{ji}} \leftarrow 0$ ;
     $t_{ji} \leftarrow t$ ;
     $k_{jt} \leftarrow k$ ;
    UpdateCounter( $S_{j,w_{ji}}, t_{ji}, k_{jt}, \text{True}$ );
     $t \leftarrow t + 1$ ;
  end
end

```

---



---

### Algorithm 3: UpdateCounter( $S_{j,w}, t, k, operation$ )

---

```

if  $operation == \text{True}$  then
  if  $t == t^{new}$  then
     $m_k \leftarrow m_k + 1$ ;
  end
  if  $S_{j,w} == 1$  then
    /* To apply GPU */;
    foreach  $(w_i, w_q) \in \mathcal{M}$  do
      if  $w == w_i$  then
         $n_{jt} \leftarrow n_{jt} + A_{i,q}$ ;
         $n_{kw} \leftarrow n_{kw} + A_{i,q}$ ;
      end
    end
  else
     $n_{jt} \leftarrow n_{jt} + 1$ ;
     $n_{kw} \leftarrow n_{kw} + 1$ ;
  end
else
  if  $S_{j,w} == 1$  then
    /* To remove counts from GPU */;
    foreach  $(w_i, w_q) \in \mathcal{M}$  do
      if  $w == w_i$  then
         $n_{jt} \leftarrow n_{jt} - A_{i,q}$ ;
         $n_{kw} \leftarrow n_{kw} - A_{i,q}$ ;
      end
    end
  else
     $n_{jt} \leftarrow n_{jt} - 1$ ;
     $n_{kw} \leftarrow n_{kw} - 1$ ;
  end
  if  $n_{jt} == 0$  then
     $m_k \leftarrow m_k - 1$ ;
  end
end

```

---

Note:  $n_{kw}$  denotes the word count of  $w$  in topic  $k$ ;  $\tilde{z}_Q$  denotes the concept related topics.