The Bulls

Digital Document Search

Yash Rathi
yrathi@buffalo.edu

Raghav Kumar
raghavku@buffalo.edu

Anoop Mathew Peringalloor
anoopmat@buffalo.edu

Yaswanth Jagilanka
yjagilan@buffalo.edu

Mariya Johar
mariyajo@buffalo.edu

Kolli Sai Nithin Reddy
sainithi@buffalo.edu

Sri Hari Gopinath Thota
sthota2@buffalo.edu

Qa

With the age of Big Data setting in and the amount of data generated is growing every day, the challenge of storing, processing, and collating data stands tall. The race to capitalize on the data among top companies and the pressure to innovate is at its forefront. The need in firms to utilize unused resources for saving time and cost is becoming an unavoidable problem. One of such domains which our project deals with is the academic field where researchers and educators have to sift through this massive data and this retrieval process can be time consuming and not cost efficient. In addition, the algorithms in place to search the documents are not designed to match the needs of the end user. Moreover, early educators and amateur learners are exploring the documents and relevant material that can fulfill their learning requirement. The redressal of these problems demands urgent need of a system or process which can provide a solution that is not only efficient but also effective. Therefore, digital document search is a way of fast retrieval which has potential of adding features such as recommendations and make it a secure retrieval. Another major position where traditional document fetching systems fail is the understanding the context behind the words. There are synonyms for each word which makes same sense, but machine might not be able to figure out the context when synonyms are used in a sentence. Therefore, these are some of the problems faced by the people in research domain and have to be resolved. The data we will be using in this problem is related to repository of research articles by variety of authors. The repository contains scholarly contributions and publications ranging from all fields ranging from computer science to physics and economics. This repository contains 1.7 million articles with relevant features such as titles, authors, categories, versions, and comments etc. Titles refers to the heading of the articles, categories contain the different subjects corresponding to the fields whereas, versions provide the different upgrades of the articles with regards to any modifications or improvement made in those articles. With this massive amount of data, we intend to create a digital document search system which can fetch the documents in a short span of time. First, we will be storing the articles in a database in Hadoop ecosystem and will be creating a dictionary of indexes related to each word. On the other end, user will be typing in the names of the documents that it needs and then those words will be mapped with documents after performing TF-IFD indexing. Once there is a match, the results will provide the complete list of documents related to the word typed in by the user initially. Therefore, the repository of research articles is used in this project to reach the final goal of making an efficient document retrieval system.

Qb

An information retrieval system is a collection of algorithms that make it easier to find relevant data or documents based on a user's needs. It not only gives the user pertinent information, but it also keeps track of how useful shown data is based on their actions. Our data is a subset of the research publications available in the ArXiv repository. We'll employ natural language processing algorithms to find the most relevant research paper for the user's inquiry. Tokenization, lemmatization, stemming, removal of stop words, word embeddings, and similarity measures such as cosine similarity (determines the semantic

distance between the user query and the retrieved content) are among the other techniques. We'd use the TF-IDF methodology, as well as topic modeling approaches like LDA, LSA, pLSA, and others.

NLP (Natural Language Processing) Operations

To make large volume of research papers into keywords to give as input for the machine learning model we are using Natural Language Processing (NLP). NLP is generally used to summarize large volumes of text. It consists of various annotations which can be used as per the requirement of the target dataset. “DocumentAssembler” is one of the operations implemented, which prepares data into a format that is processable by Spark NLP. It is used as the start of a pipeline and can read Strings or Arrays of string. Other operations include “Tokenization”, it’s a process of breaking down the given text into the smallest unit in a sentence called a token. They are mostly used to find the frequencies of the words in the entire text. “Normalizer” which removes all unwanted characters using regex pattern and transforms words as per requirement. “Lemmatizer” operation finding the base form of the related word using lexical knowledge based on the dictionary. “StopWordsCleaner” takes input string from the above operations and drops all the stop words (“are”, “a”, “the”). “Finisher” Converts the result from the above operations into a format that easier to use. Few other annotations that could be used are “Stemming”, “POSTagger”. One of the key methods of NLP is Term Frequency Inverse Document Frequency (TF-IDF). It calculates how relevant a word in a series of text. To does that by increasing proportionally to the number of times a word appears in the text and penalizes based on word frequency in the entire dataset. Few of the methods used to generate TF-IDF are “CountVectorizer” which turns text documents into vectors that include token counts information and “HashingTF” which turns documents into fixed-size vectors. The terms are mapped to indices using a Hash Function and the term frequencies are computed with respect to the mapped indices. These NLP operations are available in open-source NLP libraries like “Spark NLP”, “NLTK”, “spaCy”, “Gensim”.

NLU (Natural Language Understanding)

Natural language understanding (NLU) is a subset of natural language processing (NLP), which breaks down speech into its constituent parts to help computers understand and interpret it. While speech recognition records, transcribes, and returns text in real-time, natural language understanding (NLU) goes beyond recognition to identify a user's intent. The task of discovering topics that best characterize a set of documents is known as topic modeling. Only throughout the topic modeling process will these topics arise and that is the reason they are called latent. Latent Dirichlet Allocation (LDA) is a well-known topic modeling technique which is used to classify a topic to text in a document. It creates a Dirichlet distribution-based topic per document and word per topic model. Some of the other topics modeling techniques are LSA, which stands for Latent Semantic Analysis. The key idea is to try to decompose a matrix of documents and terms into two independent matrices which are a document-topic matrix and a matrix of topics and terms. Another technique is pLSA, which stands for Probabilistic Latent Semantic Analysis and uses a probabilistic way to solve the problem rather than the Singular Value Decomposition that we used in LSA. The key goal is to create a probabilistic model that can generate the data we see in our document-term matrix using latent or hidden themes. One more technique is BerTopic is a topic modeling technique that creates dense clusters using transformers and class-based TF-IDF. It also makes it simple to interpret and visualize the results. The program extracts document embeddings with BERT or any other embedding technique in this step. We used LDA in our project as it works better than other methods because it can generalize to new documents easily.

Query matching and Further work

In this section of query matching, initially we take the user query and perform all the preprocessing over the given query string i.e., User Query → Remove Punctuations → Remove digits if any/irrelevant → Remove Stop words → Rephrase Text → Stemming and Lemmatization → TF-IDF Calculation of the words in the user query. Then we proceed to find the cosine similarity of the TF-IDF from the user

query with the TF-IDF already derived from the documents and output the best match in this case. Also, we use another different topic modelling-based technique LDA to convert the existing document set into a bag of words, rendering every document into a specific topic. Thus, the model has been trained and based on the user query we find the closest match of the document topic with the given query. Then we ensemble the results from both techniques to find the maximum related document for the given user query. For further analysis, we are looking at Dense Passage Retrievers aka DPR, we apply a distinctive encoder EQ(·) that maps the user query to a d-dimensional vector, and retrieves set of documents of which vectors are the nearest to the query vector. We outline the similarity among the query and the passage the usage of the dot product from their vectors. Results are yet to be explore.

Qc

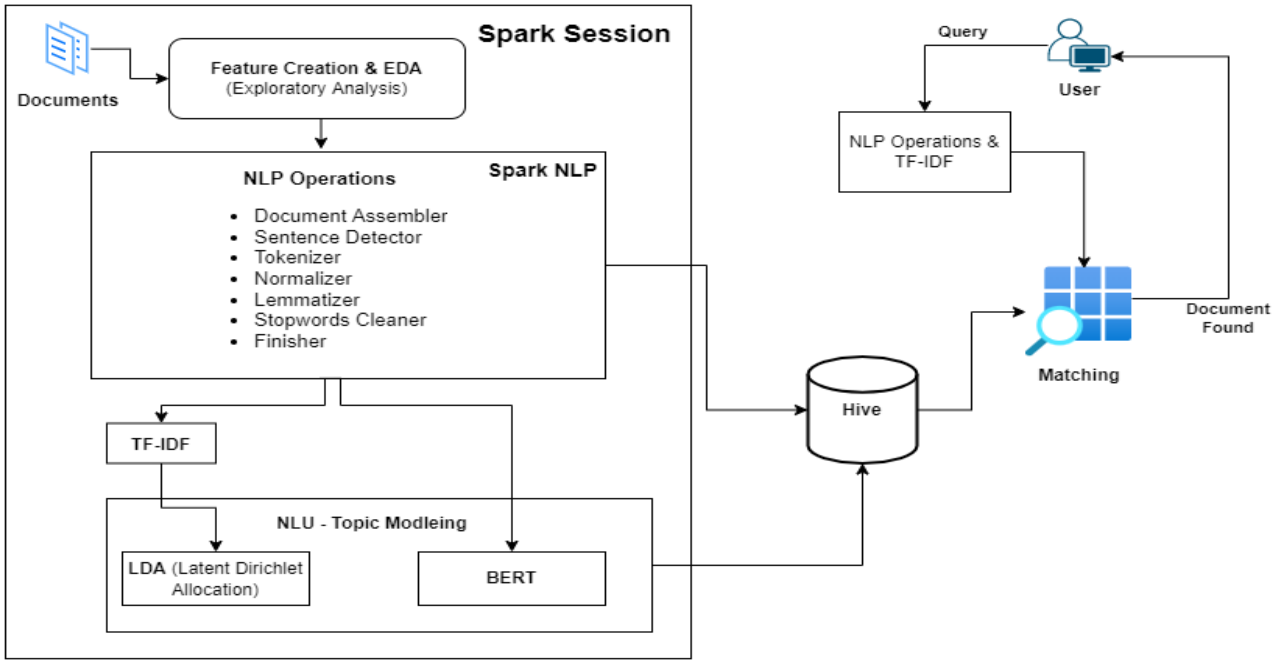
We are using the ArXiv dataset provided by Cornell University. ArXiv is a free scientific article distribution service and open-access repository with 2,040,232 articles in physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering and systems science, and economics. Since the original dataset is quite large and growing each week, we are using the machine-readable format of the ArXiv dataset which has more than 1.7 million articles present in JavaScript Object Notation. It only contains the metadata of the articles and not the entire article itself. The fields taken into consideration are id, submitter, authors, title, categories, abstract, versions, and more details about the publication, Digital Object Identifier (DOI), etc. for each article.

Our approach is given as follows: First, we will read the documents from the JSON file, next we will extract the required information from these documents like the authors, title, categories, abstract, and versions. These are the only fields that are required to perform the Document retrieval task. At this stage, we are also performing Exploratory Data Analysis (EDA) to understand the distribution of data in the dataset. The entire process is being done in Spark session. Once we have this data, we plan to perform various Natural Language Processing (NLP) techniques to clean the data like document assembler, sentence tokenization (splitting the text into sentences), removing punctuations, stop words, converting the text to lower case, word tokenization (splitting the sentence into words), Stemming and Lemmatization (converting the words into base stem words while considering the context of the word). These steps are done using Spark NLP. Once the data is cleaned, we also plan to measure Semantic relation between different words which will help us find and handle the synonymous words. After that, we will apply TF-IDF weighted indexing or similar Word embedding operation and Topic modelling methods like Bidirectional Encoder Representations from Transformers (BERT) or Latent Dirichlet Allocation (LDA), to get the most important keywords and topics from the data. This will help us classify the documents into various clusters depending on these topics and we will create a vector notation for each document using the keywords we extracted in this step. We will create a model here and save it for further use. We save these documents and their vector representations in the Hive database.

Similarly, we will take the query from the user and apply the same steps as NLP text cleaning, Word embedding, and Topic Modelling and then convert this query into a vector form. Once this is done, we will just use a similarity matching function using Cosine similarity to get the top documents based on the user query. This document is then returned to the user.

There are many existing methods to do the same task, but our modifications include implementation of Hadoop File system, semantic relation measure, and Topic modelling. The existing work has either only used Word Embeddings or Topic Modelling; we aim to implement both to get a better understanding of the data, and thus increase the precision of our model.

Below is the flowchart of our project :



Qd

For the intermediate analysis, we have completed our analysis until topic modelling. Let us understand step by step process below.

1. Creation of Spark session, data cleansing and EDA

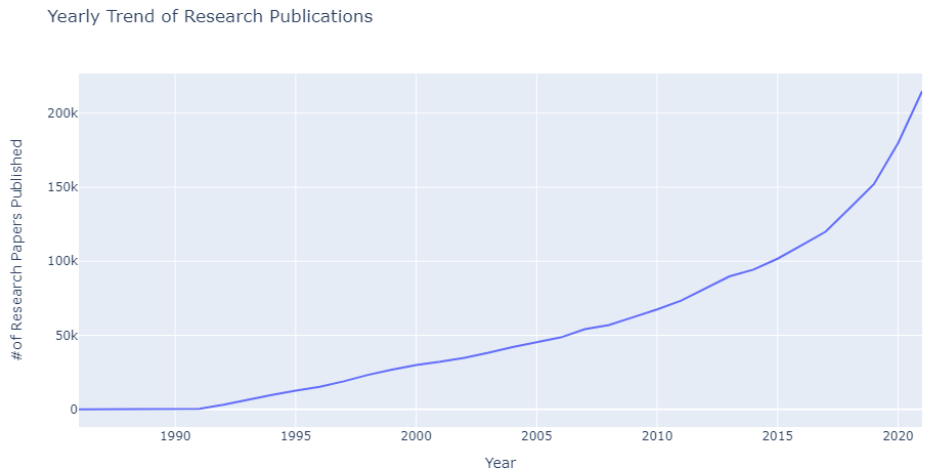
- As a first step, we have created spark session and loaded the input file which consists of information about the research papers published to ArXiv's repository. The input file captures information like id, title, abstract, authors, category etc., for each research article. Below is the screenshot of the data types of input file.

```
[('abstract', 'string'),
 ('authors', 'string'),
 ('authors_parsed', 'array<array<string>>'),
 ('categories', 'string'),
 ('comments', 'string'),
 ('doi', 'string'),
 ('id', 'string'),
 ('journal-ref', 'string'),
 ('license', 'string'),
 ('report-no', 'string'),
 ('submitter', 'string'),
 ('title', 'string'),
 ('update_date', 'string'),
 ('versions', 'array<struct<created:string,version:string>>')]
```

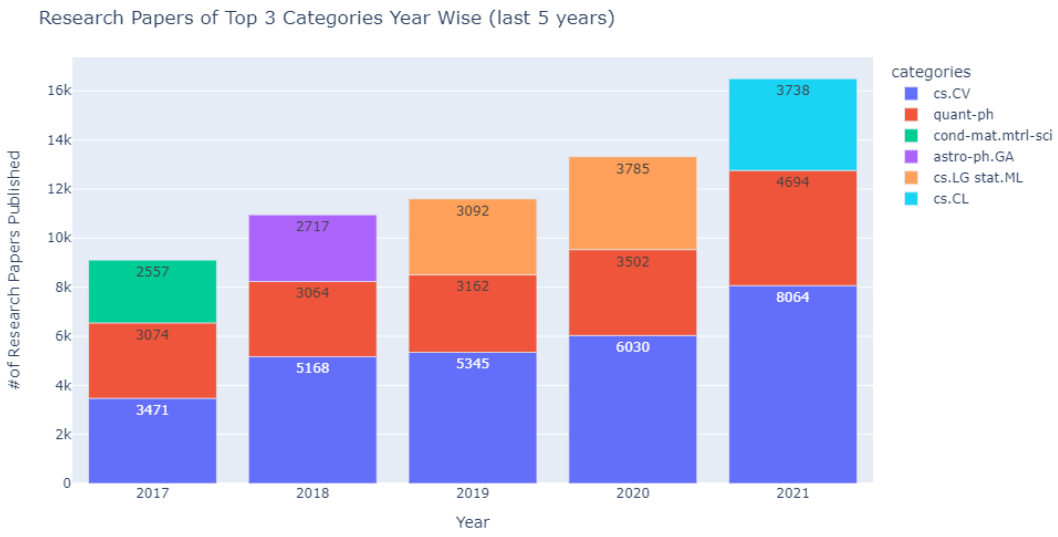
- Created new features like Year of research papers published from existing field versions, we took latest year, if we have multiple versions.

```
+---------+--------------------+--------------------+----------------+--------------------+----+--------------------+
|       id|               title|            abstract|      categories|             authors|Year|            cat_desc|
+---------+--------------------+--------------------+----------------+--------------------+----+--------------------+
|0704.0001|Calculation of pr...|  A fully differe...|          hep-ph|C. Bal\'azs, E. L...|2007|High Energy Physi...|
|0704.0002|Sparsity-certifyi...|  We describe a n...|    math.CO cs.CG|Ileana Streinu an...|2008|                null|
|0704.0003|The evolution of ...|  The evolution o...|   physics.gen-ph|         Hongjun Pan|2008|     General Physics|
|0704.0004|A determinant of ...|  We show that a ...|         math.CO|        David Callan|2007|       Combinatorics|
|0704.0005|From dyadic $\Lam...|  In this paper w...|  math.CA math.FA|Wael Abu-Shammala...|2007|                null|
|0704.0006|Bosonic character...|  We study the tw...|cond-mat.mes-hall|Y. H. Pong and C....|2007|Mesoscale and Nan...|
|0704.0007|Polymer Quantum M...|  A rather non-st...|           gr-qc|Alejandro Corichi...|2007|General Relativit...|
|0704.0008|Numerical solutio...|  A general formu...|cond-mat.mtrl-sci|     Damian C. Swift|2008|   Materials Science|
|0704.0009|The Spitzer c2d S...|  We discuss the ...|         astro-ph|Paul Harvey, Brun...|2007|        Astrophysics|
|0704.0010|Partial cubes: st...|  Partial cubes a...|         math.CO|  Sergei Ovchinnikov|2007|       Combinatorics|
|0704.0011|Computing genus 2...|  In this paper w...|  math.NT math.AG|Clifton Cunningha...|2008|                null|
|0704.0012|Distribution of i...|  Recently, Bruin...|         math.NT|         Dohoon Choi|2007|       Number Theory|
|0704.0013|$p$-adic Limit of...|  Serre obtained ...|         math.NT|Dohoon Choi and Y...|2008|       Number Theory|
|0704.0014|Iterated integral...|  In this article...|  math.CA math.AT|        Koichi Fujii|2007|                null|
|0704.0015|Fermionic superst...|  The pure spinor...|          hep-th|     Christian Stahn|2008|High Energy Physi...|
|0704.0016|Lifetime of doubl...|  In this work, w...|          hep-ph|Chao-Hsi Chang, T...|2007|High Energy Physi...|
|0704.0017|Spectroscopic Obs...|  Results from sp...|         astro-ph|Nceba Mhlahlo, Da...|2007|        Astrophysics|
|0704.0018|In quest of a gen...|  We give a presc...|          hep-th|  Andreas Gustavsson|2007|High Energy Physi...|
|0704.0019|Approximation for...|  In this note we...|  math.PR math.AG|         Norio Konno|2007|                null|
|0704.0020|Measurement of th...|  The shape of th...|          hep-ex|The BABAR Collabo...|2007|High Energy Physi...|
+---------+--------------------+--------------------+----------------+--------------------+----+--------------------+
only showing top 20 rows
```

- Performed EDA (Exploratory Data Analysis) to understand the data better.



We can observe from above chart that number of research papers published from 1990 to present almost follows exponential trend.



The above graph shows the top 5 categories by volume of research papers published over last 5 years. We can observe that **research papers in the domains of computer vision (cs.CV) and quantum physics dominate the most**.

- Other key metrics like **total number of research papers published, and total number of distinct authors are 2032539 and 1946914 respectively**.

2. NLP (Natural Language Processing) Operations using Spark NLP and TF-IDF

- We have created pipeline for NLP operations to perform on the abstract column of input file with below annotators in that order before computing TF-IDF and sending the final output from NLP operations to topic modelling.
  Document Assembler → Sentence Detector → Tokenizer → Normalizer → Lemmatizer → Stopwords Cleaner → Finisher
- Due to computational limitations, we were not able to perform TF-IDF weighting methods on the entire dataset consisting of over 2 million research papers. So, for the intermediate analysis stage we have confined our data to only those research articles which were published in last 5 years i.e., 2017 to 2021 in computer vision category within computer science.
- Term Frequency (TF) on the abstract column in the dataset is computed using CountVectorizer method. Once TF vectors are computed on entire docs, the output from TF is fed to IDF (Inverse Document Frequency) to compute the TF-IDF weighting score.
- We have then created a table to store above results on Hive.

3. NLU (Natural Language Understanding) using LDA and BERT topic modeling techniques

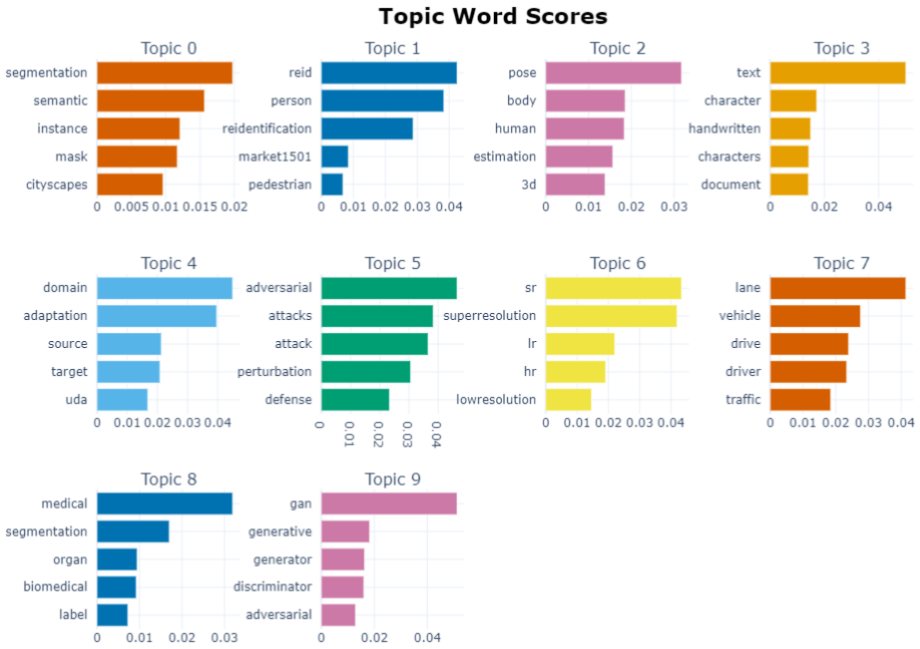
- In this section, we understood about the documents' abstracts by extracting
- The output of TF-IDF is fed to LDA (Latent Dirichlet Allocation) to extract the latent/hidden topics from abstract of each research article. Here, we are trying to compute how each document is distributed over the different topics and how each of this topic is distribution of different key words.

Top 10tTopics from LDA modelling on abstract column in the dataset

topic	words_in_topic
0	[object, feature, action, semantic, segmentation, module, model, information, video, attention]
1	[gaze, captioning, underwater, attacks, adversarial, rain, gait, attack, defense, video]
2	[text, track, object, 3d, hash, retrieval, vehicle, instance, bounding, detection]
3	[sample, data, label, learn, training, segmentation, detection, loss, feature, object]
4	[sketch, plant, detr, affine, cam, mass, subspaces, symmetric, vo, competition]
5	[segmentation, lesion, point, registration, disease, detection, patient, 3d, classification, ct]
6	[face, facial, reconstruction, 3d, domain, mesh, shape, identity, recognition, reid]
7	[crowd, point, search, cloud, count, fingerprint, clouds, emotion, density, food]
8	[light, hand, spectral, material, surface, 3d, object, denoising, map, reflectance]
9	[video, depth, scene, track, camera, pose, domain, 3d, estimation, event]

- We also performed BERT topic modelling on the abstract column after performing all the NLP operations as mentioned in above section. In this model, first the documents are embedding is done using "paraphrase-MiniLM-L12-v2" model and performs dimensionality reduction using UMAP. Finally, the clustering of UMAP is done using HDBSCAN.

Top 10 topics from BERT modeling



- The results from both the topic modelling are stored Hive database which will be used for matching user's query by cosine similarity.

User interface from where user can enter his/her query will be developed in the next phase. We will also perform cosine similarity on user's query after extracting key words with the results from TF-IDF weighting and topic modelling results and compare the accuracy.

References

https://www.analyticsvidhya.com/blog/2015/04/information-retrieval-system-explained/
https://nlp.johnsnowlabs.com/docs/en/annotators
https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05
https://hackernoon.com/nlp-tutorial-topic-modeling-in-python-with-bertopic-372w35l9
https://arxiv.org/pdf/2004.04906.pdf
https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/
https://towardsdatascience.com/nlp-extracting-the-main-topics-from-your-dataset-using-lda-in-minutes-21486f5aa925
https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf