Unsupervised Neural Categorization for Scientific Publications

Keqian Li*, Hanwen Zha*, Yu Su*, Xifeng Yan*

Abstract

Most conventional document categorization methods require a large number of documents with labeled categories for training. These methods are hard to be applied in scenarios, such as scientific publications, where training data is expensive to obtain and categories could change over years and across domains. In this work, we propose UNEC, an unsupervised representation learning model that directly categories documents without the need of labeled training data. Specifically, we develop a novel *cascade embedding* approach. We first embed *concepts*, i.e., significant phrases mined from scientific publications, into continuous vectors, which capture concept semantics. Based on the concept similarity graph built from the concept embedding, we further embed concepts into a hidden category space, where the category information of concepts becomes explicit. Finally we categorize documents by jointly considering the category attribution of their concepts. Our experimental results show that UNEC significantly outperforms several strong baselines on a number of real scientific corpora, under both automatic and manual evaluation.

1 Introduction

The large volume of scientific publications is becoming prohibitive for researchers. According to the prominent STM report [1], about 2.5 million journal articles are published in 2014 alone, and the number of publications per year is still growing at an annual rate of 3 %. Advanced techniques for better organizing, navigating, and searching scientific publications are in great demand. These techniques will not only save scientists massive amount of time, but also let outsiders quickly understand what is going on in a specific domain. A first step towards the next-generation management system for scientific publications is *document categorization*, i.e., assigning scientific publications into different categories, which provides critical information for many downstream tasks like navigation, search, and trend analysis. For example, given a set of recently published material science research articles, can we identify those related to thermal management, and then divide them into subcategories like insulation, active cooling, etc.?

Conventional document categorization methods mostly focus on general documents like news articles in a *supervised* setting, which requires a sufficient number of documents with labeled categories [2, 3]. However, manual category labeling of scientific publications could be very expensive since it can only be fulfilled by highly skilled domain experts. It will also incur a prohibitively high cost to collect labeled training data for every scientific discipline. On the other hand, some articles may come with category information. For example, the articles published in ACM conferences are often associated with category labels from the ACM classification taxonomy [4], like "*natural language processing*", which are specified by authors. However, the subject of scientific study is highly dynamic. A fixed set of categories can age quickly. The evolvement of the ACM classification taxonomy gives a clear evidence. The currently used 2012 version has changed significantly from its 1998 version: The total number of categories has increased by 90%, and only 9% of the new categories are also in the previous version, not to mention that 14 years is minuscule in the long course of scientific study. So collecting labeled training data in this way is also not sustainable.

Therefore, we propose to study the challenging setting of *unsupervised* categorization for scientific publications. Given a corpus of scientific publications (in the form of plain text documents) and a set of categories (in the form of plain text names), we aim to categorize the documents *without any labeled training data*. Free of manual labeling, unsupervised categorization brings another important benefit, that is, the freedom to specify target categories. A user can change the target categories without the cost of labeling training data for the new categories; the only cost would be to retrain the categorization model. This is critical for scientific publications because of the dynamics of scientific study.

Although few previous studies have addressed the problem of unsupervised document categorization *per se*, there are several lines of related research which can be potentially used for this problem. Topic modeling [5] extracts a set of topics, i.e., word distributions, from a text corpus, and represents each document as a distribution over topics. One can then categorize documents by *manually* associating each topic to the corresponding categories. On the other hand, one can convert unsupervised categorization into an information retrieval problem: treat each category as a keyword query, and categorize documents based on their relevance to each category query. Finally, when the target categories can be linked to some external knowledge bases, it is also possible to categorize documents in a distantly supervised fashion [6].

*University of California, Santa Barbara, {klee, hwzha, ysu, xyan}@cs.ucsb.edu

Table 1: Top concepts of each category learned by our model. Concepts are ranked by their strength of association to each category. Left: "*machine learning*" vs. "*database*". Neutral concepts that are strongly associated with neither category are also shown. Right: "*biophysics*" vs. "*optics*" vs. "*fluid dynamics*", three subfields of physics.

Category	Top Concepts
machine learning	causal models, convergence rates, decision trees, statistical inference, ensemble learning, statistical tests, support vector machines, ...
database	map and reduce, pattern mining algorithms, NNqueries, star schema, database state, transaction log, business objects, ...
neutral	strong assumption, national university of singapore, central role, edge in the graph, meta generalisation, intrinsic dimension, query nodes q, ...

Category	Top Concepts
biophysics	dna extension, rna world, viral genome, supported membranes, end monomers, transcription regulation, polymer configurations, ...
optics	inhomogeneous anisotropic, periodic layered, confocal lenslet arrays, relativistically moving, surface plasmon polariton waves, uniaxially anisotropic, ...
fluid dynamics	turbulent structures, finite elements, mechanical engineering, large eddy simulations, kelvin helmholtz instability, stagnation points, ...

We explore a different approach to this problem. Motivated by the recent development of unsupervised deep learning [7], we propose a representation learning based model, named UNEC (unsupervised neural categorization), for the problem of unsupervised categorization of scientific publications. Our key observation is that scientific publications are organized based on *concepts* that bear highly discriminative information about their categories. For example, a database paper often involves concepts like "*map and reduce*" and "*transaction*", while a machine learning paper involves concepts like "*statistical inference*" and "*convergence rate*". By leveraging state of the art methods, we can mine those concepts from the corpus and obtain a *concept representation* for each document, i.e. a document is represented based on the mined concepts, instead of single words. We then leverage neural representation learning technique to learn the strength of association of the concepts to each category (Table 1), and finally categorize each document based on its concept representation. Our model requires no external knowledge bases, which may not always exist, and no human intervention, which may be subjective. The categorization predictions are also highly interpretable because of the concept representation. It is worth noting that the proposed model can be potentially applied to general documents. We focus on scientific publications in this work because it is easy to obtain a large amount of data for automatic evaluation (see Section 5). We leave the further application on general documents to future work.

Keyphrase mining techniques [8] can be leveraged to mine significant concepts in a given corpus. The key challenge is to associate the concepts with the target categories. How can we know that the concept "*statistical inference*" has a stronger association with the category "*machine learning*" than the category "*database*", without using any labeled training data?

We propose to learn *concept embeddings* to address the aforementioned challenge. Given the concepts mined from a corpus, we embed each concept into a *category-driven* vector in a low-dimensional Euclidean space. The category attribution of concepts will become explicit in the new space, with each of the first few dimensions corresponding to a target category (Figure 1). For example, if the first dimension corresponds to "*machine learning*" and the second dimension corresponds to "*database*", then the embedding of the concept "*statistical inference*" will have a larger value on the first dimension than on the second, indicating that it has a stronger association with "*machine learning*".

However, there is a great gap from the symbolic, category-implicit concepts to their numeric, category-explicit embeddings. If labeled training data is available, one may leverage supervised classification techniques to bridge the gap. Without labeled training data, it becomes much harder. We develop a novel *cascade embedding* approach to bridge this gap. In the first stage, by leveraging word embedding techniques [9], we learn *similarity-driven* embedding of concepts, which captures the semantics as well as the similarity of concepts. Compared with the original symbolic representation, the learned concept embeddings can provide much richer information for the next stage. In the second stage, we are able to learn category-driven embedding of concepts from their similarity-drive embeddings.

The main contributions of this paper are as follows:

- We studied the novel problem of unsupervised categorization for scientific publications, without manual labeling for training data.
- We developed a novel model to address the categorization problem, where we proposed a cascade embedding approach to learn the semantics and category attribution of concepts inside the corpus, and categorize documents based on their concepts.
- We collected real datasets for the categorization task, and demonstrated the superior performance of our approach against an array of strong baseline methods.

2 Overview

We formulate the unsupervised document categorization task as follows: Given a corpus of plain text documents D, a set of target categories L, assign a *category attribution*, i.e.

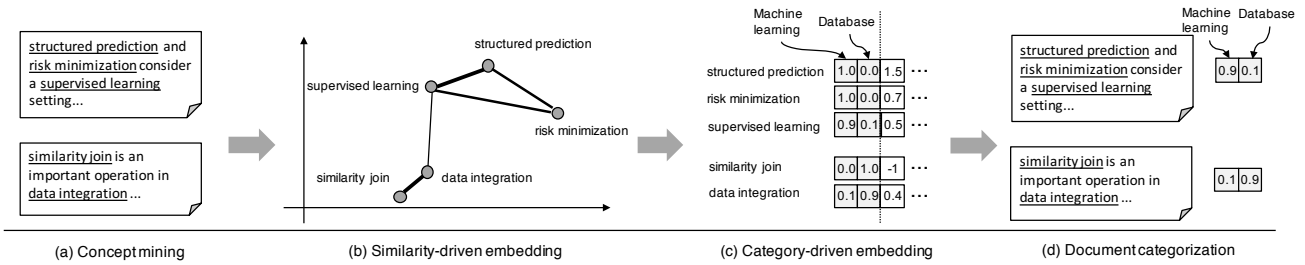


Figure 1: Pipeline of UNEC. The concept similarity graph is a complete graph, but some edges are omitted for simplicity.

a distribution over the target categories, to each document, without external knowledge bases or labeled training documents. The documents and categories are each represented as a sequence of word tokens.

Figure 1 shows the pipeline of our approach. We first mine concepts, i.e., significant phrases from the corpus (Figure 1 (a)). We propose a cascade embedding method to learn the category attribution of concepts in an unsupervised manner. The first embedding step is to learn similarity-driven embedding of concepts, which can well capture the semantic similarity between concepts, but not revealing their category attribution (Figure 1 (b)). Based on the similarity-driven concept embeddings, we build a concept similarity graph, and learn category-driven embedding of concepts with a novel regulated auto-encoder model (Figure 1 (c)). The new embedding of each concept will consist of two parts (Figure 1 (c)). The first n dimensions will be the category attribution of the concept, where n is the number of target categories, and each dimension corresponds to a single category. The rest dimensions will represent other auxiliary information of the concept. Finally, with the category attribution of concepts, it becomes straightforward to categorize a document by jointly considering all of its concepts. Next we discuss every step in details.

3 Concept Mining and Similarity-Driven Embedding

The first step in our pipeline is concept mining. A straightforward approach is to use external knowledge bases, including general-purpose ones like Wikipedia, or domain-specific ones like the ACM classification taxonomy. However, many domains may not have a well curated concept set. Even existing ones are often not complete and not updated in a timely fashion. Therefore, we propose to use a state-of-the-art key phrase mining technique, Segphrase [8], and directly use its outputted key phrase as concepts. Segphrase takes a data driven approach to mine significant phrases using corpus statistics such as popularity, concordance, informativeness, and completeness. (See Table 2) Then we obtain the "concept representation" of each document, by keeping only the concepts and remove all other words. A document is thus represented as a sequence of concepts, which is more compact but still preserves the most important information for categorization.

Given the concept representation of documents, we can apply the state of the art Skip-gram model [9] to obtain similarity-driven embedding for each concept. From the examples in Table 2, we can see that concepts having similar semantics or within similar domains will be neighbors in the embedding space. Moreover, the learned concept embeddings are beyond simple co-occurrence: the neighbors of a concept in the embedding space are usually not its neighbors in the original text, but are concepts with highly related semantics. As we will elaborate in the next section, this semantic homogeneity in the embedding space, commonly referred to as *topical similarity* [10] is particularly useful for the task of unsupervised categorization. We build a *complete* concept similarity graph (Figure 1 (b)), where nodes are the concepts, and each edge is weighted by the embedding similarity of the corresponding concept pair.

4 Category-driven Concept Embedding

In this section we propose a regularized auto-encoder model to obtain the category driven embeddings for concepts based on their similarity-driven embeddings, and use it to compute the categorization for each document.

4.1 Problem Formulation At this stage, we have mined a set of concepts C from the corpus, and associated each concept $c_i \in C$ with a similarity-driven embedding vector x_i. We denote the similarity-driven embedding space as $\mathcal{X}$. Assuming some similarity measure sim (we use cosine similarity), we can obtain a concept similarity graph. The following task is to learn the category-driven concept embedding y_i for c_i in another space $\mathcal{Y}$, where the category attribution of concepts will become explicit.

Lacking explicit supervision signal, unsupervised methods have to carefully exploit the inherent structure of the data. Although the original symbolic representation of concepts provides little useful structure, in our cascade embedding approach, the similarity-driven embedding of concepts provides much richer structure to exploit. Specifically, we focus on its topical similarity: now that we have learned semantically similar concepts such as "*generalization error*", "*leave one out error*", and "*expected risk*" are close to each

Table 2: Example concepts and their most similar neighbors mined from a collection of JMLR (Journal of Machine Learning Research) and VLDB (Very Large Data Bases) papers.

Concept	Most Similar Concepts
joint distribution	joint probability distribution, joint distributions, joint probability, conditional distribution, probability mass function, cumulative distribution
generalization error	generalisation error, generalization errors, leave one out error, generalization error bound, empirical error, true error, expected risk, training errors
bayesian network	bayesian networks, module network, bayesian network structure, structure learning algorithm, dependency network, structure learning
skyline points	skyline point, dynamic skyline, skyline layers, skyline groups, reverse skyline, skyline set, interval tree, skyline algorithms, domination tests
query plan	join operators, join orders, join predicates, input relations, query plans, plan generation, query optimizer, join plan, selection operator, outer join

other in the similarity-driven embedding space $\mathcal{X}$, can we project them into a category-driven concept embedding $\mathcal{Y}$, while preserving such proximity?

We formulate a *graph embedding* problem [11] to capture this structure. The original space $\mathcal{X}$ is represented as a concept similarity graph, and in the target space $\mathcal{Y}$ we want to preserve the pair-wise similarity in the concept similarity graph: $\forall c_i, c_j \in C$, if x_i and x_j are similar, y_i and y_j should also be similar.

In addition, we will use a data driven approach to obtain regularization signal. The idea is that, given a set of target categories, say "machine learning" and "database", there will be a lot of concepts such as "*supervised machine learning*", "*large-scale machine learning*", or "relational database", "database administrator" which we can easily tell what categories they belong to, without the need of expert human labeling or domain knowledge base. This will be the "seed concepts" for each target category. In this work, we select seed concept for each category as all concepts that contain the category name as a substring. [1]

Mathematically, for each category $cat_l, l = 1 \dots n$, we impose constraints for a set of seed concepts C_{cat_l}, so that their vectors in the target space $\mathcal{Y}$ corresponds to the correct category attribution. Starting from the seed concepts, this regularization effect will spread out over the whole concept similarity graph, and impose correct category attribution on the other concepts. More formally, we define the following guided graph embedding problem:

PROBLEM 1. (GUIDED GRAPH EMBEDDING) *Given a set of nodes (concepts) C, their similarity-driven embeddings $\{x_i | c_i \in C\}$, a similarity measure sim, a set of n target categories $\{cat_i\}_{i=1}^n$, and a set of seed concepts for each category $\{C_{cat_i}\}_{i=1}^n$, find a category-driven embedding y_i for each $x_i, c_i \in C$, that satisfy the following objective:*

$$\begin{aligned} (4.1) \qquad \textit{minimize} & \sum_{i,j \in C} ||y_i - y_j||^2 \cdot sim(x_i, x_j) \\ & + \alpha \sum_{l=1}^{n} \sum_{c_i \in C_{cat_l}} \varphi(y_i, cat_l) \\ \textit{subject to } & ||y_i|| = 1, c_i \in C, \end{aligned}$$

where α is a balance parameter for the regularization term. The norm constraint follows the convention in the original graph embedding formulation [11], with the goal to prevent from collapsing to trivial solutions.

Here the regularization term $\varphi(y_i, cat_l)$ is implemented as the cross entropy between the length n one hot vector for cat_l and the first n dimensions of y_i (i.e., the category attribution).

4.2 Solution Solving the guided graph embedding problem is a very challenging task. Because of the regularization, the analytical solution for the original graph embedding problem [11] is no longer applicable. On the other hand, because of the non-convex constraints, convex optimization methods are also not applicable, and there is no straightforward solution with optimality guarantee. Therefore, we resort to neural network methods for numerical solution.

One intuitive solution is to use a neural network to directly learn a mapping $f: \mathcal{X} \to \mathcal{Y}$ such that $y_i = f(x_i|\theta)$ using gradient descend methods following the objective in Problem 1. However, it is critical to note that the topical information, or the information about category attribution, is not embodied in the similarity-driven concept embeddings $\{x_i\}$ *per se*; rather, it is embodied in the similarities between concepts. A concept likely belongs to a category if its neighboring concepts belong to that category. The above solution would not work well because it only considers individual x_i, lacking the critical inter-concept similarity information.

Therefore, we use inter-concept similarities instead of x_i as input to the neural network. More formally, we first compute the normalized similarity matrix $S = D^{-1/2}\tilde{S}D^{-1/2}$, where $\tilde{S}_{i,j} = sim(x_i, x_j)$, and D is the degree matrix with $D_{i,i} = \sum_{c_j \in C} sim(x_i, x_j)$, 0 otherwise. We first define the following problem which aims to learn a low-dimensional matrix to best reconstruct S:

PROBLEM 2. (GUIDED GRAPH SIMILARITY RECONSTRUCTION) *Given the same inputs as Problem 1, the goal is to find matrix Y, with each row being a low-dimensional category-driven concept embedding y_i, that best reconstructs the original similarity matrix S under the Frobenius norm via certain mapping function $\Theta(Y)$, with the same regularization term*

[1] Users can also provide additional seed concepts to further improve accuracy.

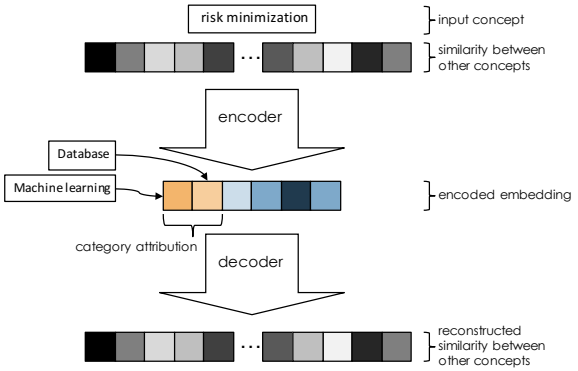


Figure 2: Illustration of the auto-encoder architecture.

$\Phi(Y)$ *and norm constraints.*

$$\underset{Y}{\textit{maximize}}\ ||S - \Theta(Y)||_F + \alpha\Phi(Y) \quad \textit{subject to}\ ||y_i|| = 1 \tag{4.2}$$

The regularization form is the same as before:

$$\Phi(Y) = \sum_{l=1}^{n} \sum_{c_i \in C_{cat_l}} \varphi(y_i, cat_l)$$

Despite the different input (i.e., similarity matrix S v.s. individual x_i), it can be proved that Problem 2 has the same optimal solution as Problem 1 under some mild condition. The detailed proof is omitted here for brevity, and can be found in supplementary materials. This provides some theoretical support for using the similarity matrix S as input instead of individual x_i. Next we present an autoencoder model to solve Problem 2, which has shown superior performance in reconstruction problems [7].

4.2.1 Auto-encoder implementation We propose a regulated auto-encoder model to learn category-driven concept embedding. It is capable of learning a representation that accurately reconstructs the original input, while being flexible enough to incorporate constraints on the neurons to account for the regularization [12].

Our model architecture is shown in Figure 2. The model consists of two components, an encoder and a decoder. The input data is the normalized similarity matrix S, with each row representing the similarity between concept i and every other concept in C. Each time, the model will take a row of the similarity matrix (denoted as s_i) as input, map it to a latent vector y_i using the encoder, and then try to reconstruct the similarity vector s_i using the decoder, denoted as $\tilde{s}_i$.

More specifically, the encoder is a multi-layer neural regressor, which is a universal approximator and is capable of learning an arbitrary mapping from s_i to y_i [13] (equation 4.3), along with a normalization layer (equation 4.4) that ensures the learned representation y_i conforms to the normalization constraints. The decoder implements the reconstruction function $\Theta(Y)$ in Problem 2. We also use a multi-layer neural regressor (equation 4.5). The regularization is imposed as a loss on the hidden representation layer as shown in equation 4.6. More formally, the prediction for a single concept $c_i \in C$ is as follows:

$$\dot{y}_i = W_2^T f(W_1^T s_i + b_1) + b_2 \tag{4.3}$$

$$y_i = \dot{y}_i / ||\dot{y}_i|| \tag{4.4}$$

$$\tilde{s}_i = \tilde{W}_1^T f(\tilde{W}_2^T f(y_i) + \tilde{b}_2) + \tilde{b}_1 \tag{4.5}$$

the objective is to minimize the overall loss

$$loss = \sum_{c_i \in C} ||\tilde{s}_i - s_i|| + \alpha\Phi(Y) \tag{4.6}$$

Here W_* and b_* are model parameters to learn; f is the activation function, for which we use the sigmoid function. $\Phi(Y)$ is the regularization term defined in equation 4.2.

Pre-training. Training the model can still be challenging because the input dimension is the same as the number of concepts, which can be large. Learning with randomly initialized parameters may be hard to converge. To overcome this problem, we propose a *pre-training* technique. That is, we first pre-compute a "nearly stable" solution $\hat{y}_i$, by analytically solving the original graph embedding problem (Problem 1 without guidance) with eigenvalue decomposition techniques [11]. Then we use the pre-computed solution to pre-train the model weights with the following loss:

$$loss_{encoder} = ||y_i - \hat{y}_i||^2 \tag{4.7}$$

$$loss_{decoder} = ||\tilde{W}_1^T f(\tilde{W}_2^T f(\hat{y}_i) + \tilde{b}_2) + \tilde{b}_1 - s_i|| \tag{4.8}$$

$$loss = loss_{encoder} + loss_{decoder} \tag{4.9}$$

These pre-trained weights are used to initialize the model.

4.3 Computing the category attribution for documents

Once we obtain the category attribution of each concept, the rest of the task is reduced to a common scoring task in information retrieval: Given a category as a query, the relevance score between each term (concept) and the query, and the containment relationship between a document and the terms, the goal is to compute the relevance score between that document and the query. We utilize the traditional TF-IDF scoring criteria [14] to compute a weight $w_{d,c}$ for a concept c to a document d, and compute the category attribution of d as $\sum_{c \in d} w_{d,c}\theta_c$, where θ_c is the category attribution of concept c.

5 Experiments

We experimentally compare the proposed method with an array of most related baseline methods, and demonstrate the superior performance of our proposed approach.

5.1 Setup Computation Environment. All the experiments were conducted on a Linux server with 12 Core(TM) i7-5930K CPU (3.50GHz), 64 GB memory, and 1 TITAN X (Pascal) GPU. The longest run of our inference algorithm described in Section 4 took less than 20 minutes to converge.

Datasets. There are two possible ways to evaluate unsupervised document categorization methods. One is to use a set of manually categorized documents, which is accurate but hard to scale. The other is to use a set of documents with automatically collected category labels, which may contain some noise but can be done at a larger scale. We will use both, but the second method will be more frequently used.

We collect documents from conference and journal proceedings, which makes it possible to automatically solicit the document categories. Our first dataset contains a complete crawl of JMLR and VLDB proceedings, resulting in a total of 3,283 papers and 31M words. For the automatic evaluation, we assign all the JMLR papers to the category "*machine learning*", and all the VLDB papers to the category "*database*".

Our second dataset contains a complete crawl of NIPS and ACL proceedings (available online), resulting in a total of 11,198 papers and 48M words. For the automatic evaluation, we assign all the NIPS papers to the category "*machine learning*", and all the ACL papers to the category "*natural language processing*". This is more challenging than the first, because historically the two research communities overlap significantly.

We also collect a third dataset of physics papers to test the ability our model outside the domain of computer science. We collect papers on arXiv under three subfields of physics, "*biophysics*", "*optics*", and "*fluid dynamics*". The resulted dataset contains 15,558 papers and 86M words. Some example concepts mined from this dataset are shown in Table 1.

In addition to automatic evaluation using the collected datasets, we also conduct a manual evaluation in two settings. The first setting is still to categorize "*machine learning*" papers against "*natural language processing*" papers using the NIPS and ACL proceedings. Because the automatically collected category labels may contain errors for these two categories, the manual evaluation may lead to more accurate evaluation of model performance. The second setting is more challenging. Instead of querying general categories like "*machine learning*", we target three more specialized sub-categories, "*bayesian learning*", "*deep learning/neural network*", and "*optimization*", and aim to find papers belonging to them. This is again conducted using the NIPS and ACL proceedings.

Evaluation Metric. We use F1 score for evaluation, a metric widely used in document classification and information retrieval literature. For each document, its category determined by a method is the one with the highest matching score. We will test all the methods in an unsupervised setting; no labeled training dataset will be provided.

5.2 Methods Compared. We compare with a wide range of related methods.

Comparative Retrieval (IR): This method treats each category as a keyword query, and scores the relevance of documents to each category via the standard TF-IDF model. The predicted category of a document is determined in a *comparative* fashion, i.e., the one with which the document has the highest relevance.

Comparative Retrieval with Query Expansion (IR+QE): We also test query expansion with word embedding [15], which works best among alternatives. It adds the k nearest neighbors (under word embedding similarity) to the original category query in the TF-IDF method. The model parameter is the number of expansion words k. We set this value by performing grid search over all possible values: from 0 to the vocabulary size.

Topic Modeling (TM): LDA [5] is a fundamental technique for modeling documents, and is still one of the most popular models used in industry. Many variants of LDA have been proposed, which either focus on improving its efficiency [16] or require supervised data [17]. Here we use the standard LDA model. We first build a topic model on the corpus via LDA, then *manually* relate the learned topics to each category according to the topic distribution of the category keywords, and finally categorize documents according to their topic distribution. We perform grid search from $[n, 10n]$ with step size n to select the number of topics, where n is the number of categories.

Dataless Classification: Dataless classification [6] is a document categorization method based on *distant supervision* [18]. Although they are distant supervision methods in nature and rely on external knowledge base like Wikipedia, we perform the comparison nonetheless. An important parameter of dataless classification is the number of Wikipedia pages to use for expanding each category. We set this value by performing grid search in the range of $[10, 30, 100, 1000]$.

Baseline with Concept Representation (X+C): We try to augment each of the above baseline methods with our concept representation, which resulted in (1) *comparative retrieval with concept expansion (IR+QE+C)*, that perform retrieval with query expansion over concepts, and (2) *concept augmented topic modeling (TM+C)*, that perform topic modeling over concepts. There is no trivial way to adapt the dataless classification method based on the way it queries the knowledge base.

PPR: We use personalized page rank (PPR) on the concept similarity graph to replace the category-driven embedding step of our method. For each node, we only keep its top 100 similar neighbors. For each category, we run personalized

page rank by setting the personalization weight of the seed concepts (see section 4) as 1, and others as 0, and get the page rank score as the category attribution.

UNEC: We evaluate our proposed method, UNEC, with the following setting. We use the default SegPhrase settings for concept extraction and learn a 200-dimensional embeddings vector of each concept via the Skip-gram model. For the auto-encoder, both the encoder and the decoder consist of 32 neurons, and the dimension of low dimension representation is 6. The most important parameter is the balance parameter for the regularization term $alpha$. We set this in a way that keeps the ratio between the regularization loss and the reconstruction loss (see Equation 4.1) to be close to 10^{-3}, as determined by a validation set of size 200. We keep this setting throughout the experiment.

5.3 Overall Performance with Automatic Evaluation

We first use the three datasets with automatically collected category labels for evaluation. The results are shown in Table 3. For the baseline methods, we report their performance with their best parameter setting. The proposed UNEC model consistently outperforms the baseline methods by a remarkable margin. The results also show that the performance of the baseline methods varies significantly across datasets. On the two datasets from the computer science domain, because the target categories are relatively easier to separate, the baseline methods are able to achieve a reasonable performance. However, on the more challenging physics dataset, the performance for many of the baseline methods degrade significantly. Part of the reason is that there are three target categories. Another important reason is that the category names are less discriminative, and it is harder to find appropriate expansion words that happen to differentiate the categories. For example, papers about "dna" may not have direct mention of word "*biophysics*". IR methods and the Dataless classification method suffer from this problem. The method IR + QE + C is an exception, showing that the mined concepts are more discriminative than general words. On the other hand, the topics identified by the topic modeling methods are not very discriminative for the target categories as well; they contain a mix of words/concepts from different categories. So topic modeling methods work poorly in this case. The baseline of PPR perform relatively well , because it is able to utilize the extracted concept and the similarity-driven embedding. The performance of UNEC is more robust, because it takes better advantage of global statistics: It learns concept semantics and categories documents by jointly considering all the concepts. UNEC consistently outperforms PPR, showing that our regulated auto-encoder model is better than PPR on this task.

5.4 Effect of Parameters

The key parameter in UNEC is the regularization weight α (Equation 4.6), which controls the balance between the reconstruction loss (*how well do the learned embeddings capture concept similarities?*) and the regularization loss (*how well do the learned embeddings respect the category constraints?*). A larger α means more weight on category regularization. Intuitively, different categorization tasks require different α values. If the categories are harder to separate, the optimal value of α shall be larger. This is supported by the experiment results shown in Figure 3. The optimal α for separating "*machine learning*" from "*database*" (JMLR vs. VLDB) is smaller than that for separating "*machine learning*" from "*natural language processing*" (NIPS vs. ACL).

However, because the role of α is to balance the two kinds of losses, we can gain more insights from the *relative regularization loss*, which is the regularization loss divided by the reconstruction loss. From the results in Figure 3, it can be observed that a good balance between the two kinds of loss is achieved when the relative regularization loss in the range of $[10^{-4}, 10^{-2}]$, i.e., the regularization loss is 2 to 4 orders of magnitude smaller than the reconstruction loss.

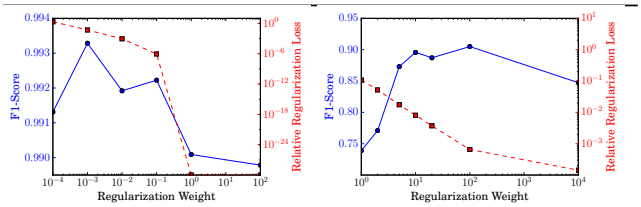
The optimal number of topics is always 2 for TM, and 4 for TM+C. Adding model capacity is not helpful in this case. For Dataless, the optimal parameter value is 30. For IR+QE, the optimal number of expansion terms is 0, while it is 100 for IR+QE+C. This shows that under the conceptualized representation similarity is better captured, so query expansion becomes more beneficial.

5.5 Qualitative Study

We show in Table 1 the top concepts for each category, ranked by the corresponding neuron activation, which would be a strongly biased "pillar" for its category. For comparison, we also show general concepts

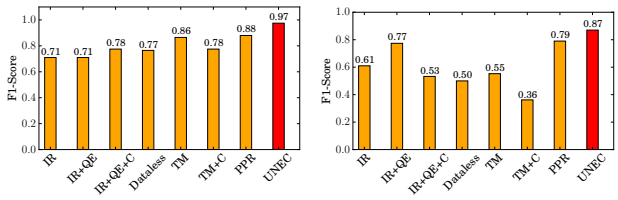
Table 3: Overall performance under automatic evaluation.

Method	JMLR vs. VLDB	NIPS vs. ACL	Physics	Avg.
IR	0.85	0.78	0.51	0.71
IR + QE	0.85	0.78	0.67	0.77
IR + QE + C	0.88	0.76	0.83	0.82
Dataless	0.83	0.78	0.68	0.76
TM	0.87	0.86	0.48	0.74
TM + C	0.76	0.77	0.37	0.63
PPR	0.90	0.81	0.87	0.86
UNEC	**0.99**	**0.91**	**0.88**	**0.93**



(a) JMLR vs. VLDB (b) NIPS vs. ACL

Figure 3: Performance of UNEC and the relative regularization loss under different regularization weight α.



(a) (b)

Figure 4: Results on manual evaluation. (a) "*machine learning*" vs. "*natural language processing*." (b) "*bayesian learning*" vs. "*deep learning/neural network*" vs. "*optimization*."

that are indifferent to the categories. Such mined concepts can potentially be used for other tasks like identifying emerging techniques.

5.6 Evaluation with human labeled ground truth Next we experiment with manually labeled testing sets. In the first experiment, the target categories are still "*machine learning*" vs. "*natural language processing*", and the corpus consists of all the NIPS and ACL papers. We employ 10 graduate students to manually categorize randomly sampled papers into the target categories, until we get 200 papers with ground truth labels. Any documents with label disagreement are discarded to ensure the label quality. All the methods are trained using the entire corpus and then tested on the manually labeled testing set. Other experiment settings are the same as before. The results are shown in Figure 4a(b). Still, UNEC significantly outperforms all the baseline methods.

We then experiment with a more challenging setting, targeting an array of more specialized categories, "*bayesian learning*," vs. "*deep learning/neural network*," vs. "*optimization*," and try to find papers belonging to these categories from the NIPS and ACL proceedings. Similar as before, we collect 200 papers, discarding any documents with label disagreement. Other settings remain the same. The results are shown in Figure 4b(b). We observe that the performance of many baseline methods degrade because of the implicitness of the category names. For example, a paper on "*Gaussian random field*", which belongs to the category of "*bayesian learning*", may not contain any direct mention of words like "*bayesian*". Because the categories are more fine-grained than before, it becomes harder to categorize concepts, e.g., to tell whether a concept belongs to "*bayesian learning*" or "*deep learning*". Methods like IR+QE+C and TM+C suffer from this problem. Jointly considering all the concepts, UNEC can still correctly categorize concepts, and achieve a good performance under this challenging setting.

6 Related work

Document categorization is a general problem studied in the field of library science, information science, and computer science [2]. Techniques for automatic text categorization have evolved from rule-based expert system to machine learning (ML) paradigm. Most of the ML based document categorization approaches are supervised in nature and apply supervised learning models to texts [3, 19]. Knowledge bases are also explored to represent the meaning of texts and perform categorization [6, 20]. These approaches cannot directly solve the unsupervised categorization problem where there is little labeled training data and knowledge base coverage.

Recent advances in information extraction, including mining concepts from text [8] or from structured data [21], concept type and relation inference and knowledge base integration [22], along with more traditional natural language processing techniques such as named entity recognition and linking [23], strongly support us to adopt the "concept represent" of each document and perform categorization more effectively.

Meanwhile, the rapid development in representation learning [7] help facilitated deeper understanding of these mined concepts. One major outcome of representation learning is a vector representation of objects that reveals their semantic meaning. Since the success of the word embedding approaches [9], the embedding learning scheme has been applied to a wide range of tasks. For example, sentence embedding [24] is proposed to embed each sentence into a vector space, which can effectively reveal its inner structure such as word importance and help relevance prediction. Network embedding [25, 26] aims to embed network vertices into vectors to capture the network structure, and help improve downstream tasks like link prediction. A central theme of representation learning is to discover a low-dimensional representation that compresses the information stored in the original input. The auto-encoder approach aims to learn such a representation using a neural network [7]. Our proposed cascade embedding approach is based upon these studies.

7 Conclusions

In this work, we studied the problem of categorizing scientific publications in a fully unsupervised setting. We employed state-of-the-art concept extraction technique to discover concepts from text corpora, utilized word embedding techniques to learn the embedding of the concepts, and proposed an auto-encoder model that extract category attribution from the learned concept embeddings, which is then used to categorize documents. We extensively evaluated our method against several carefully designed baseline methods, and demonstrated that our method significantly outperforms those strong baselines. Our research raised a series of new questions. A particularly interesting one is how to robustly handle the scenario where the number of target categories is large and categorization model will become more complex and harder to train. Techniques for encouraging model

sparsity, and the separability of concepts and documents into categories could be explored. Given the effectiveness of the concept representation, we're also interested in extending its category predicting capability to other tasks of document analysis. How can we learn a representation that most effectively reveals the semantics of a document? Can we build a more general-purpose document understanding mechanism based on the concept semantics? These are all promising directions to explore in the future.

8 Acknowledgements

The authors would like to thank the anonymous reviewers for their thoughtful comments, and Jiaming Shen, Jingbo Shang and Jiawei Han for the help with Segphrase. This research was sponsored in part by the Army Research Laboratory under cooperative agreements W911NF09-2-0053. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

References

[1] M. Ware and M. Mabe, "The STM report: An overview of scientific and scholarly journal publishing," 2015.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[3] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*. Springer, 2012, pp. 163–222.

[4] ACM classification taxonomy 2012, https://www.acm.org/publications/class-2012.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[6] M.-W. Chang, L.-A. Ratinov, D. Roth, and V. Srikumar, "Importance of semantic representation: Dataless classification." in *AAAI*, vol. 2, 2008, pp. 830–835.

[7] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[8] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1729–1744.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2013.

[10] O. Levy and Y. Goldberg, "Dependency-based word embeddings." in *ACL (2)*. Citeseer, 2014, pp. 302–308.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[12] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[13] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14, pp. 2627–2636, 1998.

[14] C. D. Manning, P. Raghavan, and H. Schütze, "Scoring, term weighting and the vector space model," *Introduction to information retrieval*, vol. 100, pp. 2–4, 2008.

[15] D. Roy, D. Paul, M. Mitra, and U. Garain, "Using word embeddings for automatic query expansion," *arXiv preprint arXiv:1606.07608*, 2016.

[16] L. Yut, C. Zhang, Y. Shao, and B. Cui, "Lda*: a robust and large-scale topic modeling system," *VLDB*, 2017.

[17] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *NIPS*, 2008.

[18] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[19] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of NAACL-HLT*, 2016, pp. 1480–1489.

[20] Y. Song and D. Roth, "On dataless hierarchical text classification." in *AAAI*, vol. 7, 2014.

[21] K. Li, Y. He, and K. Ganjam, "Discovering enterprise concepts using spreadsheet tables," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1873–1882.

[22] X. Ren, A. El-Kishky, H. Ji, and J. Han, "Automatic entity recognition and typing in massive text data," in *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2016, pp. 2235–2239.

[23] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[24] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 4, pp. 694–707, 2016.

[25] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 701–710.

[26] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1105–1114.