https://github.com/fangbo-tao/doc2cube

Doc2Cube: Allocating Documents to Text Cube without Labeled Data

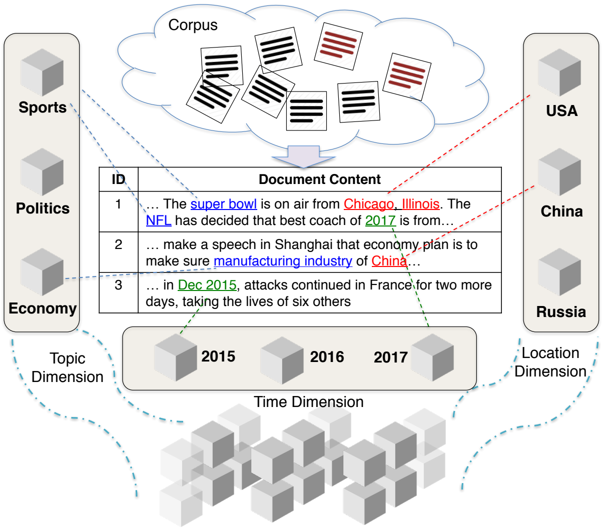
Fangbo Tao[1*], Chao Zhang[1†], Xiusi Chen[†], Meng Jiang[‡], Tim Hanratty[§], Lance Kaplan[§] and Jiawei Han[†]
[*]Facebook Inc., Menlo Park, CA, USA, Email: fangbo@fb.com
[†]University of Illinois at Urbana-Champaign, Urbana, IL, USA, Email: {czhang82, xiusic, hanj}@illinois.edu
[‡]University of Notre Dame, Notre Dame, IN, USA, Email: mjiang2@nd.edu
[§]U.S. Army Research Laboratory, Adelphi, MD, USA, Email: {timothy.p.hanratty.civ, lance.m.kaplan.civ}@mail.mil

***Abstract*—Data cube is a cornerstone architecture in multidimensional analysis of structured datasets. It is highly desirable to conduct multidimensional analysis on text corpora with cube structures for various text-intensive applications in healthcare, business intelligence, and social media analysis. However, one bottleneck to constructing text cube is to *automatically* put millions of documents into the right cube cells so that quality multidimensional analysis can be conducted afterwards—it is too expensive to allocate documents manually or rely on massively labeled data. We propose Doc2Cube, a method that constructs a text cube from a given text corpus in *an unsupervised way*. Initially, only the label names (e.g., USA, China) of each dimension (e.g., location) are provided instead of any labeled data. Doc2Cube leverages label names as weak supervision signals and iteratively performs joint embedding of labels, terms, and documents to uncover their semantic similarities. To generate joint embeddings that are discriminative for cube construction, Doc2Cube learns dimension-tailored document representations by selectively focusing on terms that are highly label-indicative in each dimension. Furthermore, Doc2Cube alleviates label sparsity by propagating the information from label names to other terms and enriching the labeled term set. Our experiments on real data demonstrate the superiority of Doc2Cube over existing methods.**

***Index Terms*—data cube, text classification, multidimensional analysis**

I. Introduction

Text cube is a multidimensional data structure with text documents residing in, where the dimensions correspond to multiple aspects (*e.g.*, topic, time, location) of the corpus. Text cube analysis has been demonstrated as a powerful text analytics tool for a wide spectrum of applications in bioinformatics, healthcare, and business intelligence. For example, by organizing a news corpus into a three-dimensional *topic-time-location* cube, decision makers can easily browse the corpus and retrieve desired articles with simple queries (*e.g.*, ⟨Sports, 2017, USA⟩). Any text mining primitives, *e.g.*, sentiment analysis, can be further applied on the retrieved data for gaining useful insights. As another example, one can organize a corpus of biomedical research papers into a neat cube structure based on different facets (*e.g.*, disease, gene, protein). Such a text cube allows people to easily identify relevant papers in biomedical research and acquire useful information for disease treatment.



ID	Document Content
1	… The super bowl is on air from Chicago, Illinois. The NFL has decided that best coach of 2017 is from…
2	… make a speech in Shanghai that economy plan is to make sure manufacturing industry of China…
3	… in Dec 2015, attacks continued in France for two more days, taking the lives of six others

Fig. 1: Text cube construction on a news corpus with three dimensions: topic, location and time. Each document needs to be assigned with one label in each of the three dimensions.

Text cube construction, *i.e.*, which automatically constructs a text cube from a text corpus, has remained largely overlooked. Specifically, given a text corpus $\mathcal{D}$ and a pre-defined cube schema $\mathcal{C}$, the task aims to allocate the documents in $\mathcal{D}$ into the right cells in $\mathcal{C}$. Figure 1 shows an example on a news corpus. Let $\mathcal{C}$ be a pre-defined cube schema with three dimensions: topic, location, and time. The text cube construction task is to assign each news article in the given corpus into a proper cube cell (*e.g.*, ⟨Sports, 2017, USA⟩), by choosing one label along each dimension to best match the textual content of the article.

Text cube construction is a multidimensional categorization problem in nature and closely related to document classification [1], [16], [17], [20]. However, the success of prevailing document classification methods largely relies on sufficient labeled document-label pairs to train reliable classifiers. For text cube construction, it is costly to manually annotate a large number of documents for classification, given that every document has to be assigned with multiple labels.

We propose Doc2Cube, a method that constructs text cube from a given text corpus in an unsupervised way. Regarding label names as a small set of labeled seed terms, Doc2Cube

[1] The first two authors have equal contributions.

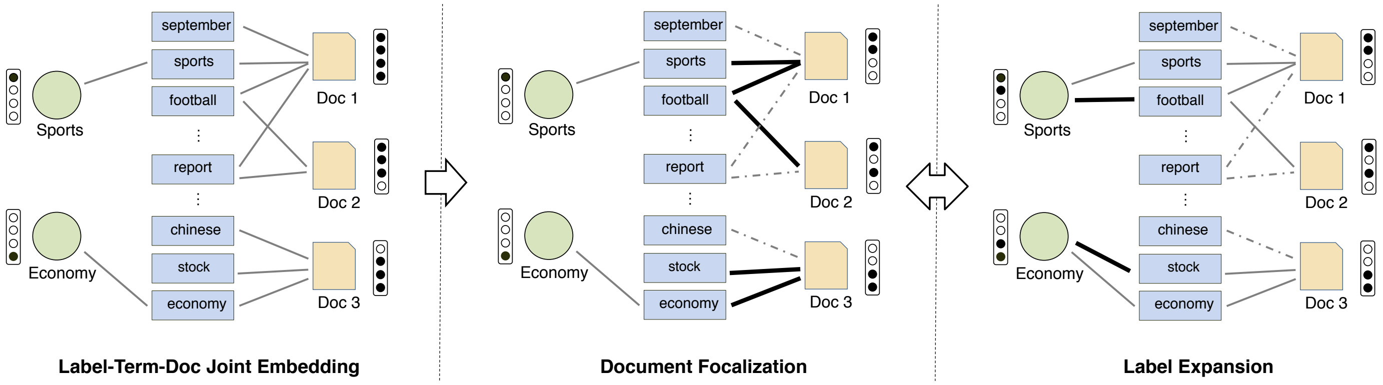


Fig. 2: A toy example of dimension-aware joint embedding framework on the *topic* dimension. In document focalization, the background term ("report") along with the indiscriminative words ("september" and "chinese") are less emphasized for the *topic* dimension. In label expansion, more topic-indicative words ("football" and "stock") are expanded and labeled.

first constructs a tripartite graph to encode the correlations among labels, terms, and documents. It then iteratively refines the graph structure and derives quality embeddings of labels, terms, and documents to uncover their inter-type similarities. During the iterative embedding process, DOC2CUBE features two components to obtain discriminative joint embeddings: *document focalization* and *label expansion*.

The document focalization component gradually sparsifies the term-document sub-graph by emphasizing discriminative terms. As shown in Figure 2, a document is initially connected with all the terms appearing in it. The resultant document embedding is *over-represented* in the sense that many terms indiscriminative to the current dimension are encoded. DOC2CUBE iteratively estimates the discriminativeness of terms for each cube dimension. As such, one document can have multiple representations—each tailored for one cube dimension by highlighting truly discriminative information.

The label expansion component iteratively densifies the label-term subgraph to address the label sparsity problem. As shown in Figure 2, as each label is only connected to its surface name in the beginning, the initial label embedding is *under-represented* because many other relevant terms are overlooked. To tackle this issue, DOC2CUBE computes the correlations between labels and terms along different dimensions, and iteratively links each label with positively correlated terms. In this way, the information is propagated from label names to other semantically relevant terms for alleviating label sparsity.

Our contributions can be summarized as follows:

1) We propose an unsupervised method for text cube construction. It does not require any labeled data, but simply leverages the surface names of different labels to achieve effective text categorization.
2) We propose a novel dimension-aware joint embedding algorithm. It learns dimension-aware embeddings by focusing on discriminative terms and propagating information from label names to other terms to alleviate label sparsity.
3) We have performed extensive experiments using two real-life datasets. The results show that our method significantly outperforms state-of-the-art methods.

II. Related Work

Lin *et al.* [11] were the first to propose the text cube concept. They assumed the text documents have been organized in a neat multidimensional structure and studied how to efficiently compute different aggregation measures in the multidimensional space. Since then, text cube analysis has drawn much attention from the database and data mining communities [4], [5], [14], [19], [21]. For example, R-Cube [14] was proposed where users can specify an analysis portion by supplying some keywords and a set of cells are extracted based on relevance. TopCell was proposed [4] to support keyword-based ranking of text cube cells and facilitate interactive exploration. However, all these studies focus on the text analytics tasks, assuming the cube is already constructed by data providers. The text cube construction task, which aims at organizing massive text documents into a cube, has remained largely overlooked.

Text cube construction is closely related to text categorization. Prevailing text categorization methods take a supervised approach. They learn reliable classifiers that are capable of predicting the label of any new document, including SVM [8], decision tree [1], [16], and neural networks [20]. Different from supervised text classification, our problem does not have any labeled data. Such a setting makes it challenging and existing supervised methods inapplicable.

There have unsupervised or weakly-supervised approaches for text categorization. Ko *et al.* [9] used heuristic rules to generate training data, but the curated labels often need considerable feature engineering efforts to ensure the quality. OHLDA [3], [6] applies topic model with given labels to generate document classifiers, while leveraging external knowledge from *Wikipedia* to represent labels. The recently developed dataless classification methods [17] also use Wikipedia to perform explicit semantic analysis of labels and documents to derive vector representations. The common limitation of OHLDA and dataless models is their dependency on external knowledge bases. They suffer from limited performance if the given corpus is closed-domain or has limited coverage by external knowledge bases.

III. Problem Definition

Given a text corpus $\mathcal{D}$, the text cube for $\mathcal{D}$ is a multidimensional data structure. Each document $d \in \mathcal{D}$ lies in one multidimensional cube cell to characterize the textual content of the document from multiple aspects. Formally, we define the concepts of *text cube* as follows:

Definition 1 (Text Cube): A text cube for a text corpus $\mathcal{D}$ is a n-dimensional structure $\mathcal{C} = (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n)$, where $\mathcal{L}_i$ is the i-th cube dimension. Each document $d \in \mathcal{D}$ resides in a cube cell $(l_{t_1}, \dots, l_{t_n})$ in $\mathcal{C}$, where l_{t_i} is the label of d in dimension $\mathcal{L}_i$.

Problem: We study the problem of constructing a text cube $\mathcal{C}$ from a text corpus $\mathcal{D}$. Let $\mathcal{C}$ be a n-dimensional text cube with dimensions $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n$, and $\mathcal{D}$ be a corpus of text documents. For any document $d \in \mathcal{D}$, text cube construction aims to allocate d into a n-dimensional cell in $\mathcal{C}$. This is equivalent to assigning n labels $l_{t_1}, \dots, l_{t_n}$ for d, where label $l_{t_i} \in \mathcal{L}_i$ represents the category of d in dimension $\mathcal{L}_i$.

IV. An Overview of Our Method

The major challenge for applying document classification methods is that there are no labeled documents for training reliable classifiers. Instead, one needs to perform document categorization along different dimensions using only label names and document content. Our method Doc2Cube uses label names to form a small set of seed labeled terms, and use them as weak supervision signals for document categorization.

Graph Embedding. As shown in Figure 2, Doc2Cube initially constructs a tripartite *label-term-document* graph to encode the relationships among labels, terms, and documents. The correlation graph for a dimension $\mathcal{L}$ is a tripartite graph $G_{LTD} = (V_{LTD}, E_{LTD})$. The node set V_{LTD} contains all the labels in $\mathcal{L}$, terms in $\mathcal{T}$, and documents in $\mathcal{D}$. The edge set E_{LTD} consists of two types of edges: (1) E_{TL} is a set of edges between labels and terms. There is an edge between term t_i and label l_j if and only if they strictly match each other, and the weight $w^{TL}_{i,j}$ is set to 1; (2) E_{TD} is a set of edges between terms and documents. There is an edge between term t_i and document d_j if t_i occurs in d_j, and the edge weight $w^{TD}_{i,j}$ is set to $\log(1 + \text{count}(t_i, d_j))$.

The graph G_{LTD} encodes the information from seed terms as well as the co-occurrence relationships between terms and documents. Based on the constructed graph, we embed all the labels, terms, and documents into a D-dimensional vector space by applying existing graph embedding techniques [18].

Dimension-Aware Updating. While the initial embeddings encode the seed information and the occurrences of terms in documents, they suffer from two drawbacks: (1) the document embeddings are *over-represented* in the sense that many terms indiscriminative to the current dimension are encoded; and (2) the label embeddings are *under-represented* because many other relevant terms are overlooked. To address the above challenges, Doc2Cube features two novel components for learning discriminative joint embeddings in an iterative fashion: (1) the *document focalization* component that emphasizes different terms for different dimensions, thus deriving dimension-aware document representations; and (2) the *label expansion* component that propagates information from label names to other terms for alleviating label scarcity. In the following section, we describe the details of these two components.

V. Learning Dimension-Aware Embeddings

In this section, we present the dimension-aware embedding updating step. Taking the joint embeddings as initialization, the updating step iteratively derives dimension-aware document embeddings by focusing on discriminative terms for each dimension, and expands the initial labeled seed terms to address label sparsity.

A. Measuring Term Discriminativeness

The key to tackling *over-representativeness* of documents and *under-representativeness* of labels is to estimate each term's discriminative power *w.r.t.* a dimension and a label. The computed discriminative scores can address document over-representativeness by emphasizing discriminative terms and understating indiscriminative ones; and meanwhile address label under-representativeness by expanding each label to highly relevant terms. In what follows, we define the *label-focal score* and the *dimension-focal score* of a term t and describe how we compute them.

1) Label-Focal Score: The label-focal score of a term t *w.r.t.* a label l in dimension $\mathcal{L}$, denoted as $f(t, l)$, aims at quantifying the discriminative power of the term t for the label l. The higher $f(t, l)$ is, the more exclusively the term t belongs to the label l. Our strategy for measuring the label-focal score $f(t, l)$ is to leverage the documents containing t to derive the distribution of term t over all the labels in dimension $\mathcal{L}$. Specifically, with the document embedding matrix $\mathbb{U}^{\mathcal{D}}$ and the label embedding matrix $\mathbb{U}^{\mathcal{L}}$, we first compute the label-document similarity matrix as:

$$\mathbf{R}^{(\mathcal{D}\mathcal{L})} = \mathbb{U}^{\mathcal{D}} \mathbb{U}^{\mathcal{L}^{\mathrm{T}}}. \qquad (1)$$

In the above, $\mathbf{R}^{(\mathcal{D}\mathcal{L})}$ is a $|\mathcal{D}| \times |\mathcal{L}|$ matrix that gives the similarities between documents and labels in the embedding space. Combining it with the term-document subgraph, we are able to further compute the similarities between labels and terms. Specifically, let $\mathbf{A}^{(\mathcal{T}\mathcal{D})}$ be the adjacency matrix for the term-document subgraph in G_{LTD}, we compute the term-label similarities as:

$$\mathbf{R}^{(\mathcal{T}\mathcal{L})} = \mathbf{A}^{(\mathcal{T}\mathcal{D})} \mathbf{R}^{(\mathcal{D}\mathcal{L})}, \qquad (2)$$

where $\mathbf{R}^{(\mathcal{T}\mathcal{L})}$ is a $|\mathcal{T}| \times |\mathcal{L}|$ matrix keeping the similarities between terms and labels. Base on $\mathbf{R}^{(\mathcal{T}\mathcal{L})}$, we apply row-wise softmax function to derive the probability distribution of each term over the labels. Finally, we define the *label-focal score* $f(t_i, l_j)$ as the probability of assigning term t_i to label l_j. Namely,

$$f(t_i, l_j) = \mathbf{R}^{(\mathcal{T}\mathcal{L})}_{ij}. \qquad (3)$$

2) Dimension-Focal Score: We proceed to define the dimension-focal score of a term. The dimension-focal score of a term t_i *w.r.t.* dimension $\mathcal{L}$, denoted as $f(t_i, \mathcal{L})$, aims to quantify how discriminative the term t_i is for the categorization task along dimension $\mathcal{L}$. The higher $f(t_i, \mathcal{L})$ is, the more useful term t_i is for deciding the label in dimension $\mathcal{L}$.

We measure the dimension-focal score $f(t_i, \mathcal{L})$ based on the distribution of term t_i over all the labels in dimension $\mathcal{L}$. Recall that the matrix $\mathbf{R}^{(\mathcal{TL})}$ gives the label distribution of term t_i. We compute its normalized KL-divergence from the uniform distribution of t_i over all the labels as the dimension-focal score. Formally, the *dimension-focal score* $f(t_i, \mathcal{L})$ is given by:

$$f(t_i, \mathcal{L}) = \frac{\sum_{j=0,\cdots,|\mathcal{L}|} \mathbf{R}_{ij}^{(\mathcal{TL})} \log |\mathcal{L}| \mathbf{R}_{ij}^{(\mathcal{TL})}}{\log |\mathcal{L}|}, \tag{4}$$

where $\log |\mathcal{L}|$ is a normalization term.

B. Document Focalization

The *document focalization* component uses the dimension-focal scores of terms to address the *over-represented* problem of document embeddings. To obtain dimension-tailored document embeddings, we use the dimension-focal scores to re-weigh the term-document matrix $\mathbf{A}^{(\mathcal{TD})}$, and compute the weighted average of term embeddings. Formally, we update the document embedding matrix $\mathbb{U}^{\mathcal{D}}$ as:

$$\mathbb{U}^{\mathcal{D}} = \left(\mathbf{A}^{(\mathcal{TD})} \circ \left[f_{\mathcal{L}} \cdots f_{\mathcal{L}} \right]_{|\mathcal{T}| \times |\mathcal{D}|} \right)^{\mathrm{T}} \mathbb{U}^{\mathcal{T}} \tag{5}$$

where $\circ$ is the Hadamard product between two matrices; and $f_{\mathcal{L}}$ is a length-$|\mathcal{T}|$ vector representing the dimension-focal scores of all the terms along dimension $\mathcal{L}$. In this formula, the dimension-focal score of each term places a penalty in the range of $[0, 1]$ on the original weight in the matrix $\mathbf{A}^{(\mathcal{TD})}$. The document embedding is then an aggregation of term embeddings with penalized weights. The higher a term's dimension-focal score is, the more it is emphasized when computing the document embedding.

C. Label Expansion

The *label expansion* component is designed to solve the *under-represented* problem of label embeddings, by linking each label with other positively correlated terms. To ensure the quality of the expanded terms, we consider two factors: (1) the label-focal score of a term; and (2) the popularity of a term. The label-focal score is critical to determining the correlations between a term and the considered label. However, we observe that only using the label-focal score could link the label to many low-quality terms during the label expansion process. This is because many terms that have high discriminative power are infrequent in the corpus. Expanding labels to them not only covers few extra documents, but also suffers from their inadequately-trained embeddings. Hence, we design the expansion criterion by combining the label-focal score and the term popularity. Given a term t_i and a label l_j, we compute the expansion score of term t_i for label l_j as:

$$e(t_i, l_j) = f(t_i, l_j) \cdot \frac{\log 1 + df(t_i)}{\log 1 + |\mathcal{D}|} > \eta \tag{6}$$

where $df(t_i)$ is the document frequency of term t_i. The second term thus reflects the normalized popularity of term t_i. In Equation 6, $\eta > 0$ is a pre-defined threshold for label expansion. Any term-label pairs with the expansion scores higher than η are connected and the adjacency matrix $\mathbf{A}^{(\mathcal{LT})}$ is updated accordingly. After the expansion, we compute the label embedding as:

$$\mathbb{U}^{\mathcal{L}} = \mathbf{A}^{(\mathcal{LT})} \mathbb{U}^{\mathcal{T}}. \tag{7}$$

Since the label expansion process changes label embeddings, the label-focal scores of terms will be updated according to the newly computed $\mathbf{R}^{(\mathcal{DL})}$ and $\mathbf{R}^{(\mathcal{TL})}$. As label-focal scores are updated, a new label expansion operation could further benefit generating high-quality label embeddings.

D. The Overall Algorithm

Algorithm 1 presents the iterative embedding updating process for document and label embeddings. Starting with the initial embeddings for labels ($\mathbb{U}^{\mathcal{L}}$), terms ($\mathbb{U}^{\mathcal{T}}$), and documents ($\mathbb{U}^{\mathcal{D}}$), we iteratively perform document focalization and label expansion to obtain more discriminative dimension-aware embeddings. In the document focalization component (lines 2 - 5), we compute the dimension-focal scores of terms, and update the document embeddings according to Equation 5; while in the label expansion component (lines 6 - 8), we compute the label-focal scores of terms, and update the label embeddings according to Equation 7. Finally, we assign the max-scoring label to each document for the target dimension. The label assignment step is achieved by directly measuring the cosine similarity between label embedding and document embedding.

VI. Experiments

A. Experimental Setup

1) Dataset: We use two datasets in our experiments: (1) The first dataset, named NYT, is a collection of New York Times articles. We crawled 13,080 articles using New York Time API in 2015. The articles in the corpus cover 29 topics and 5 countries, and each article contains exactly one topic label and one country label. (2) Our second dataset, Yelp, is a collection of business reviews from the Yelp Data Challenge with two dimensions for each review: business category and created location. Due to the long-tail nature of the raw dataset, we preprocess it by selecting the five most popular categories and states to form the label spaces, and choose the reviews falling in those top five categories and states.

Algorithm 1: Dimension-Aware Embedding Updating.

```
Input: $\mathbb{U}^{\mathcal{L}}$, $\mathbb{U}^{\mathcal{D}}$, $\mathbb{U}^{\mathcal{T}}$: initial embeddings of labels, docs and terms.
       $\mathbf{A}^{(\mathcal{LT})}$: the adjacency matrix for the label-term subgraph
       $\mathbf{A}^{(\mathcal{TD})}$: the adjacency matrix for the term-document subgraph.
       $T$: the number of iterations for updating
Output: The updated embeddings of labels and documents.
for iter = 1 : $T$ do
    // Document focalization
    Compute $\mathbf{R}^{(\mathcal{TL})}$ by Equation 1 and 2;
    for $t_i$ in $\mathcal{T}$ do
        $f(t_i, \mathcal{L}) = \frac{\sum_{j=0,\cdots,|\mathcal{L}|} \mathbf{R}_{ij}^{(\mathcal{TL})} \log |\mathcal{L}| \mathbf{R}_{ij}^{(\mathcal{TL})}}{\log |\mathcal{L}|}$
    // Update document embeddings
    $\mathbb{U}^{\mathcal{D}} = \left( \mathbf{A}^{(\mathcal{TD})} \circ \left[ f_{\mathcal{L}} \cdots f_{\mathcal{L}} \right]_{|\mathcal{T}| \times |\mathcal{D}|} \right)^{T} \mathbb{U}^{\mathcal{T}}$;
    // Label expansion
    Compute $e(t, l)$ for all term-label pairs by Equation 6;
    Update $\mathbf{A}^{(\mathcal{LT})}$ for all $e(t, l) > \eta$;
    // Update label embeddings
    $\mathbb{U}^{\mathcal{L}} = \mathbf{A}^{(\mathcal{LT})} \mathbb{U}^{\mathcal{T}}$
```

2) Baselines: We compare DOC2CUBE with multiple baselines that can perform document categorization in an unsupervised or semi-supervised way: (1) **IR** [15] treats each label as a keyword query and performs categorization based on the BM25 retrieval model. (2) **IR + Expansion (IR+QE)** extends the IR method by expanding label names using *Word2Vec* [13] and using the expanded term set as queries. (3) **Word2vec (W2V)** [13] first learns vector representations for all the terms in a given corpus, and then derives label and document representations by aggregating their member terms. Finally, the most similar label for a document is assigned based on cosine similarity. (4) **Word2vec + Focalization (W2V+DF)** extends W2V with document focalization. Instead of simply aggregating term embeddings, it leverages term dimension-focal scores to compute document representations. (5) **Paragraph2vec (P2V)** [10] directly learns vector representations of documents, by embedding documents and terms into the same semantic space. (6) **Semi-Supervised Topic Model (SEMI-TM)** [12] extends PLSA [7] by using labels as guidance and forcing the learned topics to align with the provided labels. (7) **Dataless Classification (DATALESS)** [2] is an unsupervised algorithm that utilizes Wikipedia as external knowledge base. It leverages Wikipedia and Explicit Semantic Analysis (ESA) to derive vector representations of labels and documents. (8) **PTE** [18] is a semi-supervised method that jointly embeds documents, terms, and labels into the same latent space and directly uses the embeddings for categorization.

Besides the above baseline methods, we also design two ablation algorithms of DOC2CUBE: (1) **D2C-DF** is an ablation without the label expansion component; (2) **D2C-LE** is an ablation without document focalization.

We set the parameters of different methods as follows. There are three major parameters in DOC2CUBE: (1) the latent embedding dimension D; (2) the number of iterations for embedding updating T; and (3) the correlation threshold for label expansion η. After tuning, we set these parameters as the following on both datasetes: $D = 100, T = 3$ and $\eta = 0.8$. For baselines, we set the embedding dimensions for W2V and PTE to 100 to ensure fair comparison with DOC2CUBE; we set the number of topics to 20 for SEMI-TM; and we set the number of Wikipedia concepts to 500 for DATALESS.

B. Performance Comparison

Table I reports the micro-F1 and macro-F1 scores of all the methods along different dimensions. One can observe that DOC2CUBE outperforms all the baselines in both dimensions on NYT and Yelp. Specifically, SEMI-TM is the strongest baseline along the topic and location dimensions on NYT. However, DOC2CUBE outperforms SEMI-TM by more than 16.2% in the topic dimension and 37.3% in the location dimension. On the Yelp dataset, DOC2CUBE again outperforms the strongest baseline (W2V+DF and SEMI-TM) by 22.4% and 4.5% along the business category and the location dimensions, respectively.

Comparing with the two ablations, we can observe the benefits of document focalization and label expansion. For example, on the NYT dataset, the inclusion of document focalization (D2C-DF v.s. PTE) improves the micro-F1 score from ∼0.69 to ∼0.78 in the topic dimension; and the inclusion of label expansion (D2C-LE v.s. PTE) improves the micro-F1 score from ∼0.69 to ∼0.73. Interestingly, by applying document focalization (W2V+DF) and label expansion (IR+QE) on baseline methods, we also observed considerable performance gains along different dimensions. Such a phenomenon further demonstrates the effectiveness of document focalization and label expansion.

C. Case Study

We first examine the computed dimension-focal scores of different terms on the NYT dataset. For this purpose, we pick five terms in the vocabulary and show their dimension-focal scores in the topic and location dimensions in Table II. From the results, we can see that: (1) The first two terms, "economic growth" and "soccer", both have very high focal scores in the topic dimension but low scores in the location dimension. This is intuitive as these two terms are quite topic-indicative but do not naturally reflect the location of a given article. (2) The terms "beijing" and "new york state" are only discriminative for the location dimension. (3) There are also terms that have high focal scores in both the topic and location dimensions, such as "chinese consumer". It makes sense as one can easily tell the topics and locations of news articles from such terms.

We proceed to demonstrate the empirical results of the label expansion component in Table III. Starting from the surface name of a label, DOC2CUBE is capable of discovering other terms that are highly correlated with the label. For example, for the label "movies" in the topic dimension, DOC2CUBE iteratively discovers correlated terms such as "films", "director", and "hollywood".

TABLE I: The performance of different methods on the NYT and Yelp datasets. For each dimension, we measure the micro-F1 and macro-F1 scores of different methods for categorization.

	NYT				Yelp			
	Topic		Location		Business Category		Location	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
IR	0.3963	0.4520	0.4615	0.517	0.2957	0.3669	0.0547	0.3111
IR+QE	0.4112	0.4744	0.4722	0.4726	0.3276	0.3726	0.0779	0.2806
W2V	0.5928	0.3891	0.5226	0.3598	0.4980	0.4635	0.1915	0.2530
W2V+DF	0.6100	0.3981	0.5446	0.4156	0.5129	0.5257	0.2392	0.2532
P2V	0.6079	0.4018	0.3337	0.3511	0.1920	0.3752	0.1766	0.2421
Dataless	0.5882	0.3724	0.5	0.4362	0.1463	0.1733	0.1080	0.1981
Semi-TM	0.6845	0.5407	0.5704	0.4588	0.2105	0.1876	0.3645	0.1990
PTE	0.6938	0.4992	0.595	0.4695	0.4459	0.4387	0.2505	0.2465
D2C-DF	0.7863	0.5235	0.6208	0.5635	0.6059	0.5707	0.3508	0.3010
D2C-LE	0.7347	0.5081	0.6619	0.5415	0.5261	0.5164	0.2939	0.2894
Doc2Cube	**0.7957**	**0.5414**	**0.6828**	**0.5986**	**0.6279**	**0.6037**	**0.3811**	**0.3165**

TABLE II: The dimension-focal scores of different terms in the topic and location dimension on NYT.

	Topic	Location
economic growth	0.972	0.223
soccer	0.883	0.096
beijing	0.245	0.681
new york state	0.166	0.788
chinese consumer	0.999	0.994

TABLE III: The label expansion results for four example labels in the topic and location dimensions on the NYT dataset.

Round	**Topic**			
Seed	*movies*	*baseball*	*tennis*	*business*
#1	films	inning	wimbledon	company
#2	director	hits	french open	chief executive
#3	hollywood	pitch	grand slam	industry
Round	**Location**			
Seed	*brazil*	*Australia*	*China*	*Spain*
#1	brazilian	sydney	chinese	madrid
#2	san paulo	australian	shanghai	barcelona
#3	confederations cup	melbourne	beijing	la liga

VII. Conclusion

We proposed a novel method that automatically constructs a text cube from a text corpus to facilitate multidimensional text analytics. Our proposed method, Doc2Cube, requires only the label names for document allocation. It leverages label names as weak supervision signals and iteratively performs joint embedding of labels, terms, and documents to uncover their semantic similarities. Our experiments validate the effectiveness of Doc2Cube and its advantages over a comprehensive set of baseline methods.

Acknowledgements: Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026.

References

[1] C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. *Mining text data*, pages 163–222, 2012.

[2] M. Chang, L. Ratinov, D. Roth, and V. Srikumar. Importance of semantic representation: Dataless classification. In *AAAI*, pages 830–835, 2008.

[3] X. Chen, Y. Xia, P. Jin, and J. A. Carroll. Dataless text classification with descriptive LDA. In *AAAI*, pages 2224–2231, 2015.

[4] B. Ding, B. Zhao, C. X. Lin, J. Han, and C. Zhai. Topcells: Keyword-based search of top-k aggregated documents in text cube. In *ICDE*, pages 381–384, 2010.

[5] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang. Streamcube: hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream. In *ICDE*, pages 1561–1572, 2015.

[6] V. Ha-Thuc and J. Renders. Large-scale hierarchical text classification without labelled data. In *WSDM*, pages 685–694, 2011.

[7] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, pages 289–296, 1999.

[8] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, 1999.

[9] Y. Ko and J. Seo. Automatic text categorization by unsupervised learning. In *COLING*, pages 453–459, 2000.

[10] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.

[11] C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao. Text cube: Computing IR measures for multidimensional text database analysis. In *ICDM*, pages 905–910, 2008.

[12] Y. Lu and C. Zhai. Opinion integration through semi-supervised topic modeling. In *WWW*, pages 121–130, 2008.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[14] J. M. Pérez-Martínez, R. Berlanga-Llavori, M. J. Aramburu-Cabo, and T. B. Pedersen. Contextualizing data warehouses with documents. *Decision Support Systems*, 45(1):77–94, 2008.

[15] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC*, pages 109–126, 1994.

[16] F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys*, 34(1):1–47, 2002.

[17] Y. Song and D. Roth. On dataless hierarchical text classification. In *AAAI*, pages 1579–1585, 2014.

[18] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, pages 1165–1174. ACM, 2015.

[19] F. Tao, H. Zhuang, C. W. Yu, Q. Wang, T. Cassidy, L. R. Kaplan, C. R. Voss, and J. Han. Multi-dimensional, phrase-based summarization in text cubes. *IEEE Data Eng. Bull.*, 39(3):74–84, 2016.

[20] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy. Hierarchical attention networks for document classification. In *HLT-NAACL*, pages 1480–1489, 2016.

[21] D. Zhang, C. Zhai, and J. Han. Topic cube: Topic modeling for OLAP on multidimensional text databases. In *SDM*, pages 1124–1135, 2009.