Explainable and Sparse Representations of Academic Articles for Knowledge Exploration

Keng-Te Liao[†*], **Zhihong Shen**[§], **Chiyuan Huang**[§], **Chieh-Han Wu**[§],
PoChun Chen[†*], **Kuansan Wang**[§], **Shou-de Lin**[†]
[†]National Taiwan University, [§]Microsoft Research
d05922001@ntu.edu.tw, {zhihosh,chiyuan.huang}@microsoft.com,
chiewu@microsoft.com, b02902019@ntu.edu.tw,
kuansanw@microsoft.com, sdlin@csie.ntu.edu.tw

Abstract

We focus on a recently deployed system built for summarizing academic articles by concept tagging. The system has shown great coverage and high accuracy of concept identification which could be contributed by the knowledge acquired from millions of publications. Provided with the interpretable concepts and knowledge encoded in a pre-trained neural model, we investigate whether the tagged concepts can be applied to a broader class of applications. We propose transforming the tagged concepts into sparse vectors as representations of academic documents. The effectiveness of the representations is analyzed theoretically by a proposed framework. We also empirically show that the representations can have advantages on academic topic discovery and paper recommendation. On these applications, we reveal that the knowledge encoded in the tagging system can be effectively utilized and can help infer additional features from data with limited information.

1 Introduction

Efficiently exploring knowledge is an active research topic in this era. In this work, we focus on a deployed system (Shen et al., 2018) which is built for summarizing academic publications via explainable concepts. An example[1] of the system output is shown in Figure 1, where the system tags relevant concepts such as "Word2vec" and "Word order" for summarizing the given paper. Notably, the concepts are called fields-of-study (FoS).

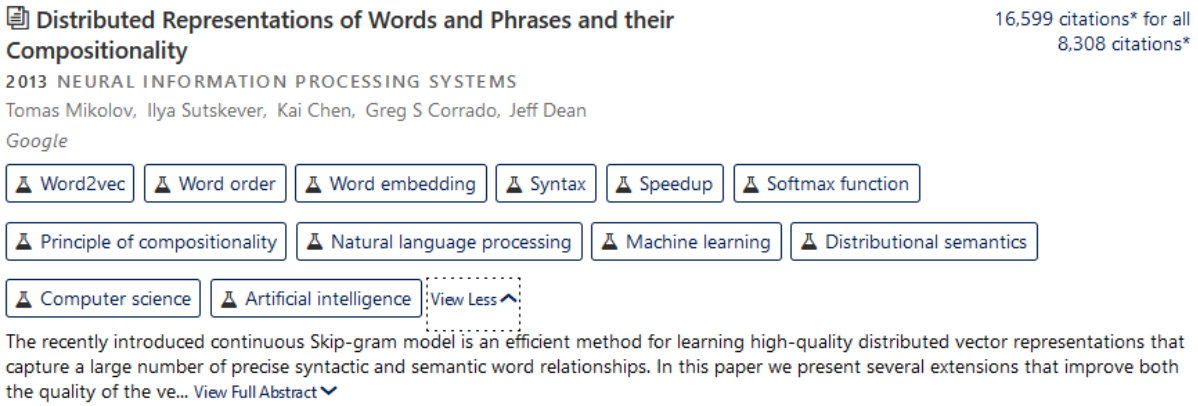

Figure 1: An example of concept (or FoS) tagging.

To ensure concepts covered by various articles can be recognized, the system acquired knowledge from 170 million academic publications via deep learning models. The learned concepts are further organized by a hierarchical structure. Specifically, the hierarchy is a 6-level tree. The FoS in level 0 are the most

[*] Work done during internship at Microsoft Research.

[1] Search result on https://academic.microsoft.com/home

Proceedings of the 28th International Conference on Computational Linguistics, pages 6207–6216
Barcelona, Spain (Online), December 8-13, 2020

coarse-grained concepts such as "Computer Science" and "Chemistry". On the contrary, FoS in level 5 are the most fine-grained concepts such as "Convolutional Deep Belief Networks" and "Phosphatase". In the tagging process, the system would normally tag both high-level and low-level FoS for helping users efficiently and thoroughly understand ideas of the given articles.

The accuracy of tagging has been carefully evaluated by human judge. However, whether the tagged FoS can be beneficial to broader applications is still unknown. We propose that this problem is worth studying as the tagging system natively possesses two advantages. The first one is the guaranteed interpretability. It potentially helps models leveraging the identified concepts be more explainable. Secondly, the tagging system has learned abundant knowledge from millions of academic papers. Ideally, the learned knowledge can be utilized for solving diverse natural language processing (NLP) problems as long as academic data are involved. In this work, we reveal that the tagged FoS can be transformed into sparse and explainable representations. We then theoretically and empirically show that the proposed representations indeed inherit the two advantages mentioned above.

This paper is organized as follows. In Section 2, we first describe how we obtain sparse representations via the FoS tagging system with slight modifications. Afterwards, we provide theoretical analyses on the representations in Section 3 by modeling the tagging system as a framework with a replaceable neural model. In the analyses, we reveal that the tagging process leveraging concept hierarchy potentially helps the sparse vectors capture more thorough semantics for measuring document similarities.

In Section 4 and 5, we evaluate representations via two applications related to knowledge exploration and can benefit from the FoS-based methods. The first one is document clustering for topic discovery. It could help users quickly classify and identify documents in a massive text corpus. We empirically show that the proposed representations and similarity measurements can gather documents with common topics more accurately. We also leverage the interpretability of FoS to explain the generated clusters.

Another application is document ranking for paper recommendation. As the queries for paper searching are usually short, we demonstrate that our tagging system is able to tag abundant FoS on extremely short documents. Particularly, the tagged FoS could have similar effects to query expansion. We then show that the FoS sparse representations can constantly help improve ranking performance especially on searching by short queries.

2 FoS Representations and Similarity

2.1 The FoS Tagging System

In this subsection, we introduce our implementation of the FoS tagging system proposed by Shen et al. (2018). We follow their methodology with two modifications. The first one is that our system does not leverage citation or reference information. It could have a negative impact on tagging, however, the benefit is that the input documents are no longer required to be included in the Microsoft academic graph (Sinha et al., 2015). The conclusions made in this work can thereby be applied to general NLP tasks or corpora. The other modification is that we preserve the confidence score of each tagged FoS for downstream processing. Our implementation is also released[2] and can be obtained by making a request.

To build the system, the first step is to obtain dense vector representations of FoS and words. The list of identified FoS can be accessed via Microsoft academic service[3]. In total, there are 228K FoS. We then train the 250-dimensional FoS and word embeddings using skip-gram (Mikolov et al., 2013) on the academic corpus containing 130 million titles and 80 million abstracts, which are the same settings proposed by Shen et al. (2018). As a result, we obtain 2 million word embeddings and 228K FoS embeddings in the same vector space.

The second step of building the system is to implement the tagging method. The general idea is to generate vector representations of input documents and measure the similarity between a document and each FoS vector for finding relevant FoS. As we do not leverage graph information, the document vectors become averaged word embeddings. Concretely speaking, a FoS is tagged if the cosine similarity

[2]https://docs.microsoft.com/en-us/academic-services/graph/language-similarity

[3]https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

between the averaged word and FoS embeddings is higher than a pre-determined threshold. In this work, we let the threshold be 0.

Regarding the tagging procedure, Shen et al. (2018) proposed not to measure similarity between a document and all 228K FoS. Instead, they proposed a strategy leveraging the concept hierarchy introduced in Section 1 to select 300-400 FoS as the candidates for each document beforehand. The motivation is to avoid expensive computation based on an observation that an academic article empirically covers no more than 20 FoS. To implement the strategy, we obtain the hierarchy via Microsoft academic service as well. With this hierarchy, the proposed strategy is to include all FoS in level 0 and level 1 as the candidates. For other levels, a FoS is recognized as a candidate only if it exactly appears as a word in the input document.

2.2 Sparse Representations and Similarity Measurement

With the preserved confidence scores, we regard the tagged FoS as sparse representations of documents. Namely, the representation of each input document is a 228K-dimensional sparse vector, where the non-zero terms are the confidence scores of the tagged FoS respectively. Given that the threshold of confidence scores is 0, the sparse vectors are non-negative. We denote the vanilla sparse representations by FoS-Sparse.

In a downstream task providing a collection of documents, we found applying inverse document frequency (IDF) weightings can be beneficial sometimes. Specifically, the confidence score in each dimension of a sparse vector is multiplied by $\log \frac{N}{df(i)}$, where N is the number of documents and $df(i)$ is the document frequency of the FoS corresponding to the i-th dimension. The document frequency here is the number of documents tagged with the FoS. We denote this variant of vanilla FoS representations by FoS-Sparse-IDF.

Given two FoS sparse vectors, we measure the similarity between two documents by cosine similarity. We note that normalizing the FoS sparse vectors to unit length is essential in this work. As introduced in Section 2.1, words in the given document happen to be identified FoS are included as the candidates. Under this strategy, a longer document tends to have more non-negative terms in its sparse representation. Without length normalization, the similarity measurement would have a bias towards document length rather than the underlying concepts.

3 Theoretical Analyses of FoS Similarity

3.1 Definitions and Modelling

In Section 3, we provide theoretical analyses for investigating the effectiveness of FoS sparse representations. In this subsection, we define symbols and formulate the procedure of obtaining FoS-based similarity scores as follows.

We denote the number of identified FoS and dimension of FoS dense embeddings as m and n, which are 228K and 250 in this work. The matrix composed of all FoS dense embeddings is denoted by $X \in \mathbb{R}^{n \times m}$. In order to compute cosine similarity, each column of X is normalized to unit length.

For an input document, as mentioned in Section 2.1, it is represented by averaged word embeddings in the same vector space as FoS dense embeddings. We denote a document dense embedding by $d \in \mathbb{R}^n$. Similarly, in order to compute cosine similarity, the l_2-norm of d is normalized to 1. We can then obtain the confidence scores between a document and all FoS by $X^T d \in \mathbb{R}^m$.

To model the hierarchy-based strategy transforming $X^T d$ into a sparse vector, we define a function $\mathcal{T}(z)$ called *threshold function*. $\mathcal{T}(z)$ takes an m-dimensional vector as input and sets the i-th dimensional value be 0 if z_i is below the threshold or the i-th FoS is not recognized as a candidate. Therefore, the FoS sparse representation of a input document is $f = \mathcal{T}(X^T d)$.

Let the FoS sparse representations and averaged word vectors of two arbitrary documents be f_a, f_b and d_a, d_b. The cosine similarity scores of these two representations are $\frac{1}{||f_a||_2||f_b||_2} \cdot f_a^T f_b$ and $d_a^T d_b$ respectively. In Section 3.2 and 3.3, we analyze the core of FoS-based similarity, $f_a^T f_b$, and discuss the possibility of being a more effective measurement method over $d_a^T d_b$.

3.2 Analysis of FoS Similarity Without Threshold Function

For easier explanation, we tentatively ignore the threshold function $\mathcal{T}(z)$ here. Without $\mathcal{T}(z)$, $f_a^T f_b = (X^T d_a)^T (X^T d_b) = d_a^T (XX^T) d_b = d_a^T W d_b$. One can notice that $W \in \mathbb{R}^{n \times n}$ is an empirical covariance matrix multiplying a constant $m-1$ if the mean column of X is a n-dimensional zero vector[4]. By ignoring the constant $m-1$, the element w_{ij} in W is the covariance of i-th and j-th dimensions of the FoS dense embeddings X.

Compared with $d_a^T d_b$, the quadratic form $d_a^T W d_b$ covers cross-dimensional similarities. Specifically, $d_a^T W d_b = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} d_{a,i} d_{b,j} = \sum_{i=j} w_{ij} d_{a,i} d_{b,j} + \sum_{i \neq j} w_{ij} d_{a,i} d_{b,j}$. The term $\sum_{i=j} w_{ij} d_{a,i} d_{b,j}$ is $d_a^T d_b$ with dimensional weightings. The additional term $\sum_{i \neq j} w_{ij} d_{a,i} d_{b,j}$ could be regarded as measuring similarity of d_a and d_b in distinct dimensions. As $d_{a,i} d_{b,j}$ are weighted by the corresponding covariance, the product value of two less correlated dimensions would then have smaller impact on the final similarity score $d_a^T W d_b$.

3.3 Analysis of FoS Similarity With Threshold Function

We first introduce a lemma and a theorem for discussing the effect of including the threshold function $\mathcal{T}(z)$.

Lemma 1. *Given a document a, there exists a function $\mathcal{T}_a$ such that $f_a = \mathcal{T}(X^T d_a) = \mathcal{T}_a(X^T) d_a$.*

Proof. The value in the i-th dimension of the vector $\mathcal{T}(X^T d_a)$ is either 0 or $x_i^T d_a$, where x_i is the i-th column of X. The i-th value in $\mathcal{T}(X^T d_a)$ is 0 if and only if $x_i^T d_a$ is lower than the threshold or the corresponding FoS is not selected as a candidate. To construct $\mathcal{T}_a$, we first obtain a temporary vector $Y_a = \mathcal{T}(X^T d_a)$ and let $\mathcal{T}_a$ be a function sparsifying X^T by setting x_i be a zero vector if the i-th value in Y_a is 0. Namely, we obtain the 0 values of f_a by $\mathbf{0}^T d_a$. Let the sparsified X^T be X_a^T, we then have $X_a^T d_a = \mathcal{T}_a(X^T) d_a = \mathcal{T}(X^T d_a)$.

Theorem 1. *Given two documents a and b, there exists a function $\hat{\mathcal{T}}$ such that $f_a^T f_b = \mathcal{T}(X^T d_a)^T \mathcal{T}(X^T d_b) = d_a^T \hat{\mathcal{T}}(X, X^T) d_b = d_a^T \hat{W}_{ab} d_b$, where the matrix $\hat{W}_{ab}$ is symmetric.*

Proof. By Lemma 1, $f_a^T f_b = d_a^T \mathcal{T}_a(X) \mathcal{T}_b(X^T) d_b$. We can then construct a function $\mathcal{T}_{ab}$ such that $d_a^T \mathcal{T}_a(X) \mathcal{T}_b(X^T) d_b = d_a^T \mathcal{T}_{ab}(X) \mathcal{T}_{ab}(X^T) d_b$. By definition and derivation in Lemma 1, $f_a^T f_b = \sum_{i=1}^{m} x_{a,i}^T d_a \cdot x_{b,i}^T d_b$, where $x_{a,i}^T$ and $x_{b,i}^T$ are the i-th row of X_a^T and X_b^T respectively. It can be seen that if either $x_{a,i}^T$ or $x_{b,i}^T$ is a zero vector, the term $x_{a,i}^T d_a \cdot x_{b,i}^T d_b$ will be 0. It implies that in this situation, setting both $x_{a,i}^T$ and $x_{b,i}^T$ be zero vectors will not change $f_a^T f_b$. By following the idea of proving Lemma 1, we then construct a matrix X_{ab}^T which is a sparsified X^T. The construction is done by setting the i-th row of X^T be zero vector if either the i-th value of $\mathcal{T}_a(X^T) d_a$ or $\mathcal{T}_b(X^T) d_b$ is 0. Therefore, we can have $d_a^T \mathcal{T}_a(X) \mathcal{T}_b(X^T) d_b = d_a^T X_{ab} X_{ab}^T d_b = d_a^T \mathcal{T}_{ab}(X) \mathcal{T}_{ab}(X^T) d_b = d_a^T \hat{\mathcal{T}}(X, X^T) d_b = d_a^T \hat{W}_{ab} d_b$.

By Theorem 1, it can be seen that $f_a^T f_b$ with $\mathcal{T}(z)$ still measures cross-dimensional similarity while some covariance information is discarded. The discarded information can be formulated by $d_a^T W d_b - d_a^T \hat{W}_{ab} d_b = d_a^T (W - \hat{W}_{ab}) d_b = \sum_{i=1}^{n} \sum_{j=1}^{n} (w_{ij} - \hat{w_{ij}}) d_{a,i} d_{b,j}$. Let $v_{ij} = w_{ij} - \hat{w}_{ij}$. Since the weighting v_{ij} is decided by less relevant FoS, it may inaccurately estimate the similarity. We also find that v_{ij} is usually much higher than $\hat{w}_{ij}$, making the term $v_{ij} d_{a,i} d_{b,j}$ non-negligible in $d_a^T W d_b$. Indeed, as mentioned in Section 2.1 that usually only a small number of FoS are relevant to an academic article, there would be around 228K less relevant FoS contributing to v_{ij}. In summary, the matrix W could help capture more comprehensive semantics while it simultaneously comes with strong noise. Therefore, the existence of a filtering method such as $\mathcal{T}(z)$ would be important. For further verification, empirical studies are provided in Section 5.4.

[4]It is approximately true in our system. The mean and standard deviation of the vector values are 3.7×10^{-4} and 5×10^{-4}.

4 Academic Document Clustering

4.1 Task and Dataset

In this Section, we evaluate the effectiveness of representations via academic document clustering. The documents for clustering are required to be composed of some collections with distinct topics respectively. An example would be publications collected from different research fields. We then assume that if representations are sufficiently effective, the predefined topics can be discovered unsupervisedly. If using FoS-based methods, we show an additional advantage which is the ability of summarizing a cluster by high-level concepts. Together with the generated clusters, it could help users quickly identify and understand articles from a massive corpora. We demonstrate this feature in Section 4.3 by showing the dominant level 1 (L1) FoS in each cluster.

The dataset for the clustering task is Cora (McCallum et al., 2000). It is a citation graph with 30,635 nodes where each node is a published paper with titles and abstracts. We concatenate the title and abstract, and let it be a document for each node. In addition to text data, the nodes are also labeled with hierarchical topics up to 3 levels such as */Artificial Intelligence/Machine Learning/Reinforcement Learning/*. In our experiments, we specifically take the top level topics as the labels. In total, there are 10 different labels which are "Operating Systems", "Networking", "Hardware and Architecture", "Artificial Intelligence", "Databases", "Information Retrieval", "Encryption and Compression", "Programming", "Human Computer Interaction", and "Data Structures, Algorithms and Theory".

4.2 Experimental Settings

Given that there are 10 categories of labeled topics, we generate 10 document clusters by the following methods and see which one is the most consistent with the ground truth.

Latent Dirichlet Allocation (LDA): It is the well-known method for discovering and inferencing hidden topics from a document set. We adopt gensim implementation (Řehůřek and Sojka, 2010) for experiments. To estimate the performance more accurately, we repeat the training and testing processes 30 times with different random seeds, and average the performance scores as the experiment results.

MAG skip-gram: The document representations are obtained by averaging the 250-dimensional word vectors introduced in Section 2.1. The dense document embeddings are denoted by MAG-SG. To find clusters, we apply k-means clustering.

Bag-of-words (BoW) with TF-IDF: The documents are represented by unit-length sparse vectors where the value in each dimension is the corresponding term frequency multiplying inverse document frequency. Note that the stop words are removed in the experiments. The sparse vectors are then clustered by k-means clustering.

FoS-based methods: The documents are represented by the proposed FoS sparse representations, FoS-Sparse and FoS-Sparse-IDF. The sparse vectors are normalized to unit lengths and clustered by k-means clustering.

4.3 Clustering Results

We first examine how many FoS are tagged to a Cora document by our system. By experiments, in average there are 14.89 tagged FoS for a document. The number is consistent with the empirical observation by Shen et al. (2018) that a scientific article usually covers no more than 20 concepts. We also compare the number with averaged vocabulary size of a Cora document, which is 62.43. Note that the stop words are removed beforehand. Therefore, we could firstly observe that FoS are more spatially efficient than BoW as a representation method.

Regarding the clustering performance, we conduct common clustering metrics including Adjusted Rand index (ARI), Normalized Mutual Information (NMI), Homogeneity (Rosenberg and Hirschberg, 2007) and Completeness (Rosenberg and Hirschberg, 2007) for evaluations. The results are shown in

	ARI	NMI	Homogeneity	Completeness
LDA	0.107	0.156	0.155	0.158
MAG-SG	0.060	0.108	0.112	0.104
BoW	0.080	0.191	0.191	0.192
FoS-Sparse	0.110	0.193	0.210	0.178
FoS-Sparse-IDF	**0.145**	**0.219**	**0.234**	**0.205**

Table 1: Performance of document clustering. ARI and NMI are abbreviations of Adjusted Rand index and Normalized Mutual Information respectively.

cluster 1	cluster 2	cluster 3	cluster 4
Distributed Computing	Computer Network	Parallel Computing	Artificial Intelligence
Embedded System	Distributed Computing	Computer Hardware	Pattern Recognition
Operating System	Computer Architecture	Computer Engineering	Machine Learning
cluster 5	**cluster 6**	**cluster 7**	**cluster 8**
Information Retrieval	Information Retrieval	Arithmetic	Software Engineering
Database	NLP	Computer Network	Parallel Computing
Programming Language	World Wide Web	Computer Security	PL
cluster 9	**cluster 10**		
Multimedia	Pure Mathematics		
HCI	Discrete Mathematics		
CGI	Combinatorics		

Table 2: The 3 most frequently tagged L1 FoS in the clusters generated by FoS-IDF. PL, HCI and CGI are abbreviations of Programming Language, Human Computer Interaction and Computer Graphics Images respectively.

Table 1. From the table, we can see that the FoS-based methods can result in the most consistent clusters with the ground truth. Also, the performance gaps between MAG skip-gram and FoS-based methods could support the argument in Section 3 that the cross-dimensional similarities could be implicitly captured and be beneficial.

For the two FoS-based methods, it can be seen that the IDF weightings are helpful. A possible reason could be that there exist FoS similar to stop words. For example, a FoS called "algorithm" is constantly tagged to documents in Cora. It is reasonable as the documents in Cora are all computer science papers. However, for example, the "algorithm" FoS may not be helpful for distinguishing whether the input document belongs to "Databases" or "Information Retrieval". In this case, decreasing the impact of "algorithm" could help concentrate on other potentially more important FoS.

We finally demonstrate the ability of FoS in summarizing document clusters. For each cluster generated by FoS-Sparse-IDF, we extract 3 L1 FoS which are most frequently tagged to documents in the cluster. The extracted L1 FoS are listed in Table 2. As can be seen, the core concepts of the clusters can thus be revealed and are generally consistent with the 10 categories defined by Cora.

5 Paper Recommendation

5.1 Task and Dataset

In this section, we evaluate representations via paper recommendation. The goal is to extract and rank papers in order of relevance to users queries. In the experiments, we let the measured similarity between a query and a document be the relevance score, and examine whether the resulting ranks can be consistent with the results provided by human beings.

The dataset we use is User Study dataset (Kanakia et al., 2019). It contains 2,014 academic papers, where 147 of them are query papers. For each query paper, at most 20 relevant papers are recommended by human experts. Additionally, the human experts are asked to provide a score ranging from 1 to 5 for

quantifying the relevance when recommending a paper. Currently the papers in User Study dataset does not have text content. We then crawled paper titles and abstracts from Microsoft academic graph by the provided paper IDs for experiments.

5.2 Experimental Settings

As the queries provided by users are usually a few words, we generate query data by two different methods for simulating the scenario.

Titles as queries: We take titles of the 147 query papers as the actual queries. The averaged length of the titles is 8.79.

Words randomly sampled from abstracts as queries: The averaged length of abstracts in User Study dataset is 191.78. To get compact queries, we randomly sample words from the abstract of each query paper. The sampling rate ranges from 10% to 100% with 10% interval (i.e. 10%, 20%, 30%, etc.), where 100% rate means the query is identical to the original abstract. In order to augment testing data and estimate ranking performance more accurately, for each sampling rate between 10% and 90%, we generate 100 queries with different random seeds.

With generated query data, we obtain relevance scores between queries and candidate papers by the following methods, where the content of the candidate papers are abstracts.

BM25: BM25 is a well-known and effective document retrieval method. In the experiments, we set k_1 and b in the score function of BM25 be 1.2 and 0.75 respectively.

MAG skip-gram: The queries and candidate papers are represented by averaged word embeddings with l_2-normalization. The obtained document embeddings are the same as we introduced in Section 4.2. To verify the arguments in Section 3.3, we compare two similarity measurements $d_a^T d_b$ and $d_a^T W d_b$. The two similarity measurements are denoted by MAG-SG and MAG-SG-QUAD respectively.

FoS-based methods: We tag FoS on queries and candidate papers by our system for obtaining FoS-Sparse and FoS-Sparse-IDF representations. The similarity is measured by $\frac{1}{||f_a||_2||f_b||_2} \cdot f_a^T f_b$ introduced in Section 3.1.

With the relevance scores, we rank the candidate papers for each query and evaluate the results by averaging normalized discounted cumulative gain at 5 (nDCG@5) scores.

5.3 The Tagged FoS of Queries

As tagging FoS on short documents or sentences was not investigated by Shen et al. (2018), we first check whether our system can tag reasonable FoS on short queries. Examples of tagging are provided in Table 3. In the table, the first two rows in the queries column are two paper titles and the last two rows are two abstracts with 10% sampling rate. Due to the space limit, only top 6 FoS and their confidence scores are listed.

We could firstly see that the tagged FoS can generally be relevant to the queries. Secondly, we noted that the tagged FoS could provide additional information for document retrieval. Take the first row for example, the tagged FoS "Machine Learning" can help identify documents also related to machine learning while do not contain the three words "Generate", "Adversarial", or "Nets". Therefore, the effect of FoS on short documents could be similar to query expansion, where the ability could come from the knowledge acquired from 170 million papers. We finally examine the number of tagged FoS. The averaged number of tagged FoS on titles and abstracts are 9.35 and 16.89 respectively. Compared with their original lengths, 8.79 and 191.78 in average, we could see that the tagged FoS can be compact summaries for long documents while be extended textual features for short documents.

Queries	Top 6 Tagged FoS and Confidence Scores
Generative Adversarial Nets	Generative grammar: 0.37, Adversarial system: 0.35, Artificial intelligence: 0.33, Machine Learning: 0.31 Computer Science: 0.29, NLP: 0.29
Image Watermarking With Better Resilience	Digital watermarking: 0.55, Computer vision: 0.39 Data mining: 0.39, Pattern recognition: 0.39 Machine learning: 0.39, Computer Science: 0.38
driver the cloud describe microsoft drivers our us of to present results	Cloud Computing: 0.48, Operating System: 0.35 Software Engineering: 0.35, Data Science: 0.35 Database: 0.34, Human-Computer-Interaction: 0.34
centmail limiting of protocol neither joining money begins send a no client's the large number to many account	Communication source: 0.54, Computer security: 0.43 Internet privacy: 0.42, Computer network: 0.42 World Wide Web: 0.41, Telecommunications: 0.41

Table 3: Tagged FoS of queries.

BM25	MAG-SG	MAG-SG-QUAD	FoS-Sparse	FoS-Sparse-IDF
0.688	0.635	0.605	**0.691**	0.690

Table 4: Averaged nDCG@5 scores on User Study dataset. The queries are paper titles.

5.4 Performance of Unsupervised Paper Ranking

The averaged nDCG@5 scores of quering by titles and reduced abstracts are shown in Table 4 and Figure 2. The observations and discussions are summarized as follows.

- MAG-SG, MAG-SG-QUAD and FoS-based methods respectively correspond to $d_a^T d_b$, $d_a^T W d_b$ and $d_a^T \hat{W}_{ab} d_b$ introduced in Section 3. Here we can see the advantage of $\hat{W}_{ab}$ which boosts the ranking performance. We can also see the matrix W is not directly beneficial to downstream tasks. As argued in Section 3.3, the irrelevant information could have strong and negative impacts and can be eliminated by the threshold function with hierarchy-based strategy.

- When the queries are sufficiently long, BM25 shows the best performance in the experiments. However, if the queries contain limited information, FoS-based methods can have more advantages. From Figure 2, we can see that FoS-Sparse can outperform BM25 if sampling rate is lower than 50% which is around 55 words. As discussed in Section 5.3, the advantages could come from the query expansion effects compensating missing information.

- Different from the results in Section 4.3, including the IDF weightings of FoS does improve the performance. A possible reason could be the topics covered by User Study dataset are more diverse than Cora. Therefore, there exist fewer FoS similar to stop words which are constantly observed while encode less relevant information.

5.5 Ensemble Methods

As shown that BM25 and FoS-Sparse have respective advantages, here we investigate the effects of combining the two methods. The combination method is simply the weighted sum of estimated relevance scores. For example, given a query and a candidate document, let the relevance scores obtained by BM25 and FoS-Sparse be S_{bm} and S_{fs} respectively, the relevance score estimated by the combination method is $S_{bm} + \beta S_{fs}$.

Although β could be a hyper-parameter, we attempt to investigate the effects of optimized β in this work. We do the optimization by a learning to rank method, ListNet (Cao et al., 2007), which is a listwise ranking approach. Since we only have one learnable variable, the amount of labelled data could be less demanding.

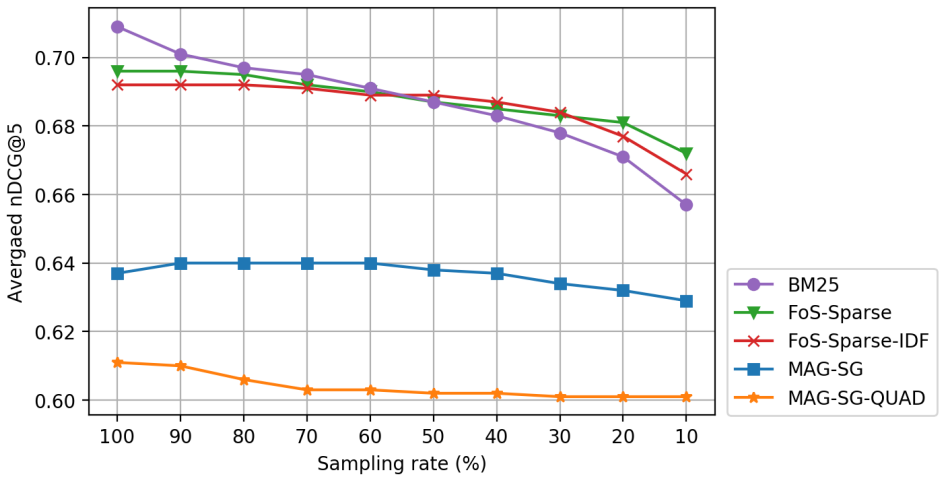

Figure 2: Averaged nDCG@5 scores on User Study dataset. The queries are words sampled from abstracts.

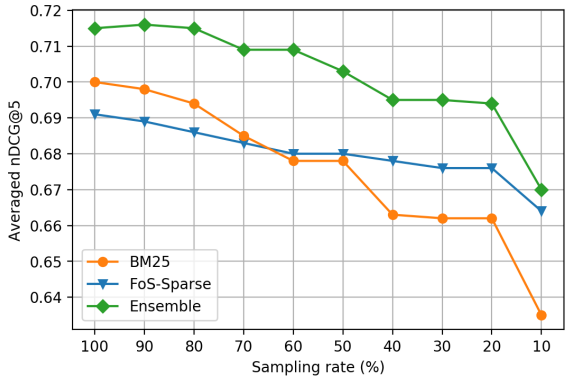

Figure 3: Averaged nDCG@5 scores on sampled User Study dataset. The queries are words sampled from abstracts.

BM25	FoS-Sparse	Ensemble
0.687	0.691	**0.705**

Table 5: Averaged nDCG@5 scores on sampled User Study dataset. The queries are paper titles.

In the experiments, we randomly sample 50% data from User Study dataset as the labelled data for training, and let the remaining 50% be testing data. We repeat the process 30 times and average the nDCG@5 scores for estimating final performance. Notably, when the queries are generated by sampling from abstracts, we train a β for each sampling rate. The averaged nDCG@5 scores are reported in Table 5 and Figure 3.

From the results, we could firstly see that BM25 and FoS-Sparse show similar behaviors reported in Section 5.4. After simple combination, the distinct advantages of the two methods can not only be retained but also be strengthened. It could support the previous argument that FoS tagging can capture features of textual data in different aspects.

6 Conclusion

In this work, we investigated whether the concepts summarizing academic publications can be effective sparse representations for diverse applications. To examine the effectiveness, we provided theoretical analyses and empirical studies for verification. We presented that the tagging method with the concept hierarchy could have an effect on capturing more thorough semantics. We also revealed several advantages of the proposed representations. The intrinsic advantages would be interpretability and compactness, helping people understand the underlying knowledge more efficiently. When coming to downstream tasks, the knowledge learned from millions of publications can show the impact. As reported,

the tagging system can infer additional features from data with limited information. Also, the additional features can be easily combined with common ranking methods and achieve better performance.

For future work, a direction would be focusing on the neural model providing the dense word and FoS representations. As the current system has a limitation that dynamic contextual information is not included, it could be interesting to investigate whether similar conclusions can be made on contextualized embeddings.

References

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136, New York, NY, USA. ACM.

Anshul Kanakia, Zhihong Shen, Darrin Eide, and Kuansan Wang. 2019. A scalable hybrid research paper recommender system for microsoft academic. In *The World Wide Web Conference*, WWW '19, pages 2893–2899, New York, NY, USA. ACM.

Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163. www.research.whizbang.com/data.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June. Association for Computational Linguistics.

Zhihong Shen, Hao Ma, and Kuansan Wang. 2018. A web-scale system for scientific knowledge exploration. In *Proceedings of ACL 2018, System Demonstrations*, pages 87–92, Melbourne, Australia, July. Association for Computational Linguistics.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*, pages 243–246.