hyperdoc2vec: Distributed Representations of Hypertext Documents

Jialong Han[♠], Yan Song[♠], Wayne Xin Zhao[♦], Shuming Shi[♠], Haisong Zhang[♠]
[♠]Tencent AI Lab
[♦]School of Information, Renmin University of China
{jialonghan,batmanfly}@gmail.com,{clksong,shumingshi,hansonzhang}@tencent.com

Abstract

Hypertext documents, such as web pages and academic papers, are of great importance in delivering information in our daily life. Although being effective on plain documents, conventional text embedding methods suffer from information loss if directly adapted to hyper-documents. In this paper, we propose a general embedding approach for hyper-documents, namely, hyperdoc2vec, along with four criteria characterizing necessary information that hyper-document embedding models should preserve. Systematic comparisons are conducted between hyperdoc2vec and several competitors on two tasks, *i.e.,* paper classification and citation recommendation, in the academic paper domain. Analyses and experiments both validate the superiority of hyperdoc2vec to other models w.r.t. the four criteria.

1 Introduction

The ubiquitous World Wide Web has boosted research interests on hypertext documents, *e.g.,* personal webpages (Lu and Getoor, 2003), Wikipedia pages (Gabrilovich and Markovitch, 2007), as well as academic papers (Sugiyama and Kan, 2010). Unlike independent plain documents, a hypertext document (hyper-doc for short) links to another hyper-doc by a hyperlink or citation mark in its textual content. Given this essential distinction, hyperlinks or citations are worth specific modeling in many tasks such as link-based classification (Lu and Getoor, 2003), web retrieval (Page et al., 1999), entity linking (Cucerzan, 2007), and citation recommendation (He et al., 2010).

To model hypertext documents, various efforts (Cohn and Hofmann, 2000; Kataria et al., 2010; Perozzi et al., 2014; Zwicklbauer et al., 2016; Wang et al., 2016) have been made to depict networks of hyper-docs as well as their content. Among potential techniques, distributed representation (Mikolov et al., 2013; Le and Mikolov, 2014) tends to be promising since its validity and effectiveness are proven for plain documents on many natural language processing (NLP) tasks.

Conventional attempts on utilizing embedding techniques in hyper-doc-related tasks generally fall into two types. The first type (Berger et al., 2017; Zwicklbauer et al., 2016) simply downcasts hyper-docs to plain documents and feeds them into word2vec (Mikolov et al., 2013) (w2v for short) or doc2vec (Le and Mikolov, 2014) (d2v for short). These approaches involve downgrading hyperlinks and inevitably omit certain information in hyper-docs. However, no previous work investigates the information loss, and how it affects the performance of such downcasting-based adaptations. The second type designs sophisticated embedding models to fulfill certain tasks, *e.g.,* citation recommendation (Huang et al., 2015b), paper classification (Wang et al., 2016), and entity linking (Yamada et al., 2016), *etc.* These models are limited to specific tasks, and it is yet unknown whether embeddings learned for those particular tasks can generalize to others. Based on the above facts, we are interested in two questions:

- What information should hyper-doc embedding models preserve, and what nice property should they possess?
- Is there a general approach to learning task-independent embeddings of hyper-docs?

To answer the two questions, we formalize the hyper-doc embedding task, and propose four criteria, *i.e., content awareness*, *context awareness*, *newcomer friendliness*, and *context intent aware-*

ness, to assess different models. Then we discuss simple downcasting-based adaptations of existing approaches w.r.t. the above criteria, and demonstrate that none of them satisfy all four. To this end, we propose hyperdoc2vec (h-d2v for short), a general embedding approach for hyper-docs. Different from most existing approaches, h-d2v learns two vectors for each hyper-doc to characterize its roles of citing others and being cited. Owning to this, h-d2v is able to directly model hyperlinks or citations without downgrading them. To evaluate the learned embeddings, we employ two tasks in the academic paper domain[1], *i.e.,* paper classification and citation recommendation. Experimental results demonstrate the superiority of h-d2v. Comparative studies and controlled experiments also confirm that h-d2v benefits from satisfying the above four criteria.

We summarize our contributions as follows:

- We propose four criteria to assess different hyper-document embedding models.
- We propose hyperdoc2vec, a general embedding approach for hyper-documents.
- We systematically conduct comparisons with competing approaches, validating the superiority of h-d2v in terms of the four criteria.

2 Related Work

Network representation learning is a related topic to ours since a collection of hyper-docs resemble a network. To embed nodes in a network, Perozzi et al. (2014) propose DeepWalk, where nodes and random walks are treated as pseudo words and texts, and fed to w2v for node vectors. Tang et al. (2015b) explicitly embed second-order proximity via the number of common neighbors of nodes. Grover and Leskovec (2016) extend DeepWalk with second-order Markovian walks. To improve classification tasks, Tu et al. (2016) explore a semi-supervised setting that accesses partial labels. Compared with these models, h-d2v learns from both documents' connections and contents while they mainly focus on network structures.

Document embedding for classification is another focused area to apply document embeddings. Le and Mikolov (2014) employ learned d2v vectors to build different text classifiers. Tang et al. (2015a) apply the method in (Tang et al., 2015b) on word co-occurrence graphs for word embeddings, and average them for document vectors. For hyper-docs, Ganguly and Pudi (2017) and Wang et al. (2016) target paper classification in unsupervised and semi-supervised settings, respectively. However, unlike h-d2v, they do not explicitly model citation contexts. Yang et al. (2015)'s approach also addresses embedding hyper-docs, but involves matrix factorization and does not scale.

Citation recommendation is a direct downstream task to evaluate embeddings learned for a certain kind of hyper-docs, *i.e.,* academic papers. In this paper we concentrate on context-aware citation recommendation (He et al., 2010). Some previous studies adopt neural models for this task. Huang et al. (2015b) propose Neural Probabilistic Model (NPM) to tackle this problem with embeddings. Their model outperforms non-embedding ones (Kataria et al., 2010; Tang and Zhang, 2009; Huang et al., 2012). Ebesu and Fang (2017) also exploit neural networks for citation recommendation, but require author information as additional input. Compared with h-d2v, these models are limited in a task-specific setting.

Embedding-based entity linking is another topic that exploits embeddings to model certain hyper-docs, *i.e.,* Wikipedia (Huang et al., 2015a; Yamada et al., 2016; Sun et al., 2015; Fang et al., 2016; He et al., 2013; Zwicklbauer et al., 2016), for entity linking (Shen et al., 2015). It resembles citation recommendation in the sense that linked entities highly depend on the contexts. Meanwhile, it requires extra steps like candidate generation, and can benefit from sophisticated techniques such as collective linking (Cucerzan, 2007).

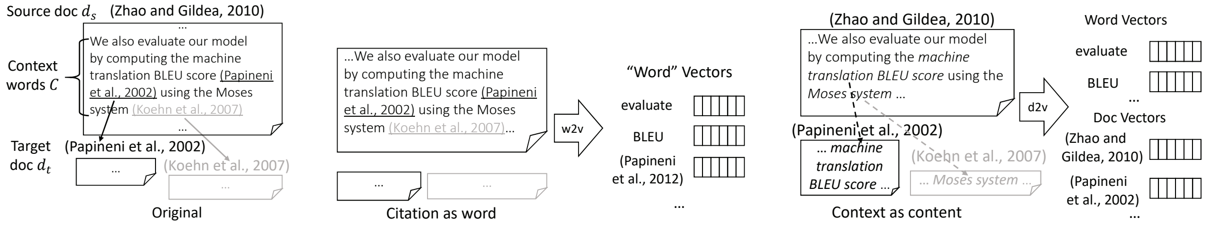
3 Preliminaries

We introduce notations and definitions, then formally define the embedding problem. We also propose four criteria for hyper-doc embedding models w.r.t their appropriateness and informativeness.

3.1 Notations and Definitions

Let $w \in W$ be a word from a vocabulary W, and $d \in D$ be a *document id* (*e.g.,* web page URLs and paper DOIs) from an id collection D. After filtering out non-textual content, a *hyper-document* H is reorganized as a sequence of words and doc ids,

[1] Although limited in tasks and domains, we expect that our embedding approach can be potentially generalized to, or serve as basis to more sophisticated methods for, similar tasks in the entity domain, *e.g.,* Wikipedia page classification and entity linking. We leave them for future work.



(a) Hyper-documents. (b) Citation as word. (c) Context as content.

Figure 1: An example of Zhao and Gildea (2010) citing Papineni et al. (2002) and existing approaches.

i.e., $W \cup D$. For example, web pages could be simplified as streams of words and URLs, and papers are actually sequences of words and cited DOIs.

If a document id d_t with some surrounding words C appear in the hyper-doc of d_s, *i.e.,* H_{d_s}, we stipulate that a *hyper-link* $\langle d_s, C, d_t \rangle$ is formed. Herein $d_s, d_t \in D$ are ids of the *source* and *target* documents, respectively; $C \subseteq W$ are *context words*. Figure 1(a) exemplifies a hyperlink.

3.2 Problem Statement

Given a corpus of hyper-docs $\{H_d\}_{d \in D}$ with D and W, we want to learn document and word embedding matrices $\mathbf{D} \in \mathbb{R}^{k \times |D|}$ and $\mathbf{W} \in \mathbb{R}^{k \times |W|}$ simultaneously. The i-th column $\mathbf{d}_i$ of $\mathbf{D}$ is a k-dimensional embedding vector for the i-th hyper-doc with id d_i. Similarly, $\mathbf{w}_j$, the j-th column of $\mathbf{W}$, is the vector for word w_j. Once embeddings for hyper-docs and words are learned, they can facilitate applications like hyper-doc classification and citation recommendation.

3.3 Criteria for Embedding Models

A reasonable model should learn how contents and hyperlinks in hyper-docs impact both $\mathbf{D}$ and $\mathbf{W}$. We propose the following criteria for models:

- **Content aware.** Content words of a hyper-doc play the main role in describing it, so the document representation should depend on its own content. For example, the words in Zhao and Gildea (2010) should affect and contribute to its embedding.

- **Context aware.** Hyperlink contexts usually provide a summary for the target document. Therefore, the target document's vector should be impacted by words that others use to summarize it, *e.g.,* paper Papineni et al. (2002) and the word "*BLEU*" in Figure 1(a).

- **Newcomer friendly.** In a hyper-document network, it is inevitable that some documents are not referred to by any hyperlink in other hyper-docs. If such "newcomers" do not get embedded properly, downstream tasks involving them are infeasible or deteriorated.

- **Context intent aware.** Words around a hyperlink, *e.g.,* "evaluate ... by" in Figure 1(a), normally indicate why the source hyper-doc makes the reference, *e.g.,* for general reference or to follow/oppose the target hyper-doc's opinion or practice. Vectors of those context words should be influenced by both documents to characterize such semantics or intents between the two documents.

We note that the first three criteria are for hyper-docs, while the last one is desired for word vectors.

4 Representing Hypertext Documents

In this section, we first give the background of two prevailing techniques, `word2vec` and `doc2vec`. Then we present two conversion approaches for hyper-documents so that `w2v` and `d2v` can be applied. Finally, we address their weaknesses w.r.t. the aforementioned four criteria, and propose our `hyperdoc2vec` model. In the remainder of this paper, when the context is clear, we mix the use of terms hyper-doc/hyperlink with paper/citation.

4.1 `word2vec` and `doc2vec`

`w2v` (Mikolov et al., 2013) has proven effective for many NLP tasks. It integrates two models, *i.e.,* `cbow` and `skip-gram`, both of which learn two types of word vectors, *i.e.,* IN and OUT vectors. `cbow` sums up IN vectors of context words and make it predictive of the current word's OUT vector. `skip-gram` uses the IN vector of the current word to predict its context words' OUT vectors.

As a straightforward extension to `w2v`, `d2v` also has two variants: `pv-dm` and `pv-dbow`. `pv-dm` works in a similar manner as `cbow`, except that the IN vector of the current document

Desired Property	Impacts Task?		Addressed by Approach?			
	Classification	Citation Recommendation	w2v	d2v-nc	d2v-cac	h-d2v
Context aware	✓	✓	✓	×	✓	✓
Content aware	✓	✓	×	✓	✓	✓
Newcomer friendly	✓	✓	×	✓	✓	✓
Context intent aware	×	✓	×	×	×	✓

Table 1: Analysis of tasks and approaches w.r.t. desired properties.

Model	Output			
	$\mathbf{D}^I$	$\mathbf{D}^O$	$\mathbf{W}^I$	$\mathbf{W}^O$
w2v	✓	✓	✓	✓
d2v (pv-dm)	✓	×	✓	✓
d2v (pv-dbow)	✓	×	×	✓
h-d2v	✓	✓	✓	✓

Table 2: Output of models.

is regarded as a special context vector to average. Analogously, pv-dbow uses IN document vector to predict its words' OUT vectors, following the same structure of skip-gram. Therefore in pv-dbow, words' IN vectors are omitted.

4.2 Adaptation of Existing Approaches

To represent hyper-docs, a straightforward strategy is to convert them into plain documents in a certain way and apply w2v and d2v. Two conversions following this strategy are illustrated below.

Citation as word. This approach is adopted by Berger et al. (2017).[2] As Figure 1(b) shows, document ids D are treated as a collection of special words. Each citation is regarded as an occurrence of the target document's special word. After applying standard word embedding methods, *e.g.,* w2v, we obtain embeddings for both ordinary words and special "words", *i.e.,* documents. In doing so, this approach allows target documents interacting with context words, thus produces context-aware embeddings for them.

Context as content. It is often observed in academic papers when citing others' work, an author briefly summarizes the cited paper in its citation context. Inspired by this, we propose a context-as-content approach as in Figure 1(c). To start, we remove all citations. Then all citation contexts of a target document d_t are copied into d_t as additional contents to make up for the lost information. Finally, d2v is applied to the augmented documents to generate document embeddings. With this approach, the generated document embeddings are both context- and content-aware.

4.3 hyperdoc2vec

Besides citation-as-word with w2v and context-as-content with d2v (denoted by d2v-cac for short), there is also an alternative using d2v on documents with citations removed (d2v-nc for short). We made a comparison of these approaches in Table 1 in terms of the four criteria stated in Section 3.3. It is observed that none of them satisfy all criteria, where the reasons are as follows.

First, w2v is not content aware. Following our examples in the academic paper domain, consider the paper (hyper-doc) Zhao and Gildea (2010) in Figure 1(a), from w2v's perspective in Figure 1(b), "...computing the machine translation BLEU ..." and other text no longer have association with Zhao and Gildea (2010), thus not contributing to its embedding. In addition, for papers being just published and having not obtained citations yet, they will not appear as special "words" in any text. This makes w2v newcomer-unfriendly, *i.e.,* unable to produce embeddings for them. Second, being trained on a corpus without citations, d2v-nc is obviously not context aware. Finally, in both w2v and d2v-cac, context words interact with the target documents without treating the source documents as backgrounds, which forces IN vectors of words with context intents, *e.g.,* "*evaluate*" and "*by*" in Figure 1(a), to simply remember the target documents, rather than capture the semantics of the citations.

The above limitations are caused by the conversions of hyper-docs where certain information in citations is lost. For a citation $\langle d_s, C, d_t \rangle$, citation-as-word only keeps the co-occurrence information between C and d_t. Context-as-content, on the other hand, mixes C with the original content of d_t. Both approaches implicitly downgrade citations $\langle d_s, C, d_t \rangle$ to $\langle C, d_t \rangle$ for adaptation purposes.

To learn hyper-doc embeddings without such limitations, we propose hyperdoc2vec. In this model, two vectors of a hyper-doc d, *i.e.,* IN and OUT vectors, are adopted to represent the document of its two roles. The IN vector $\mathbf{d}^I$ characterizes d being a source document. The OUT vector $\mathbf{d}^O$ encodes its role as a target document. We note that learning those two types of vectors is advantageous. It enables us to model citations and con-

[2] It is designed for document visualization purposes.

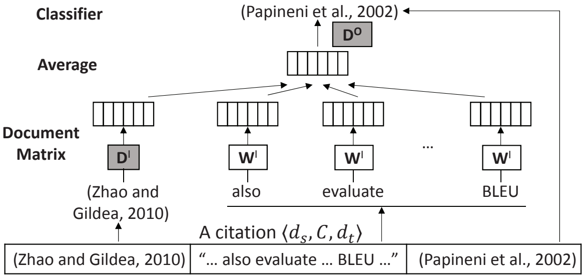


Figure 2: The `hyperdoc2vec` model.

tents simultaneously without sacrificing information on either side. Next, we describe the details of `h-d2v` in modeling citations and contents.

To model citations, we adopt the architecture in Figure 2. It is similar to `pv-dm`, except that documents rather than words are predicted at the output layer. For a citation $\langle d_s, C, d_t \rangle$, to allow context words C interacting with both vectors, we average $\mathbf{d}_s^I$ of d_s with word vectors of C, and make the resulted vector predictive of $\mathbf{d}_t^O$ of d_t. Formally, for all citations $\mathcal{C} = \{\langle d_s, C, d_t \rangle\}$, we aim to optimize the following average log probability objective:

$$\max_{\mathbf{D}^I, \mathbf{D}^O, \mathbf{W}^I} \frac{1}{|\mathcal{C}|} \sum_{\langle d_s, C, d_t \rangle \in \mathcal{C}} \log P(d_t | d_s, C) \quad (1)$$

To model the probability $P(d_t | d_s, C)$ where d_t is cited in d_s with C, we average their IN vectors

$$\mathbf{x} = \frac{1}{1 + |C|} \left(\mathbf{d}_s^I + \sum_{w \in C} \mathbf{w}^I \right) \quad (2)$$

and use $\mathbf{x}$ to compose a multi-class softmax classifier on all OUT document vectors

$$P(d_t | d_s, C) = \frac{\exp(\mathbf{x}^\top \mathbf{d}_t^O)}{\sum_{d \in D} \exp(\mathbf{x}^\top \mathbf{d}^O)} \quad (3)$$

To model contents' impact on document vectors, we simply consider an additional objective function that is identical to `pv-dm`, *i.e.,* enumerate words and contexts, and use the same input architecture as Figure 2 to predict the OUT vector of the current word. Such convenience owes to the fact that using two vectors makes the model parameters compatible with those of `pv-dm`. Note that combining the citation and content objectives leads to a joint learning framework. To facilitate easier and faster training, we adopt an alternative pre-training/fine-tuning or *retrofitting* framework (Faruqui et al., 2015). We initialize with a predefined number of `pv-dm` iterations, and then optimize Eq. 1 based on the initialization.

Dataset		Docs	Citations	Years
NIPS	Train	1,590	512	Up to 1998
	Test	150	89	1999
	Total	1,740	601	Up to 1999
ACL	Train	18,845	91,792	Up to 2012
	Test	1,563	16,937	2013
	Total	20,408	108,729	Up to 2013
DBLP	Train	593,378	2,565,625	Up to 2009
	Test	55,736	308,678	From 2010
	Total	649,114	2,874,303	All years

Table 3: The statistics of three datasets.

Finally, similar to `w2v` (Mikolov et al., 2013) and `d2v` (Le and Mikolov, 2014), to make training efficient, we adopt negative sampling:

$$\log \sigma(\mathbf{x}^\top \mathbf{d}_t^O) + \sum_{i=1}^{n} \mathbb{E}_{d_i \sim P_N(d)} \log \sigma(-\mathbf{x}^\top \mathbf{d}_i^O) \quad (4)$$

and use it to replace every $\log P(d_t | d_s, C)$. Following Huang et al. (2015b), we adopt a uniform distribution on D as the distribution $P_N(d)$.

Unlike the other models in Table 1, `h-d2v` satisfies all four criteria. We refer to the example in Figure 2 to make the points clear. First, when optimizing Eq. 1 with the instance in Figure 2, the update to $\mathbf{d}^O$ of Papineni et al. (2002) depends on $\mathbf{w}^I$ of context words such as "*BLEU*". Second, we pre-train $\mathbf{d}^I$ with contents, which makes the document embeddings content aware. Third, newcomers can depend on their contents for $\mathbf{d}^I$, and update their OUT vectors when they are sampled[3] in Eq. 4. Finally, the optimization of Eq. 1 enables mutual enhancement between vectors of hyper-docs and context intent words, *e.g.,* "*evaluate by*". Under the background of a machine translation paper Zhao and Gildea (2010), the above two words help point the citation to the BLEU paper (Papineni et al., 2002), thus updating its OUT vector. The intent "*adopting tools/algorithms*" of "evaluate by" is also better captured by iterating over many document pairs with them in between.

5 Experiments

In this section, we first introduce datasets and basic settings used to learn embeddings. We then discuss additional settings and present experimental results of the two tasks, *i.e.,* document classification and citation recommendation, respectively.

[3] Given a relatively large n.

Model	Original		w/ DeepWalk	
	Macro	Micro	Macro	Micro
DeepWalk	61.67	69.89	61.67	69.89
w2v (I)	10.83	41.84	31.06	50.93
w2v (I+O)	9.36	41.26	25.92	49.56
d2v-nc	70.62	77.86	70.64	78.06
d2v-cac	71.83	78.09	71.57	78.59
h-d2v (I)	68.81	76.33	**73.96**	**79.93**
h-d2v (I+O)	**72.89**	**78.99**	73.24	79.55

Table 4: F_1 scores on DBLP.

Model	Content Aware/ Newcomer Friendly	Original		w/ DeepWalk	
		Macro	Micro	Macro	Micro
DeepWalk	-	66.57	**76.56**	66.57	76.56
w2v (I)	× / ×	19.77	47.32	59.80	72.90
w2v (I+O)	× / ×	15.97	45.66	50.77	70.08
d2v-nc	✓/ ✓	61.54	73.73	69.37	78.22
d2v-cac	✓/ ✓	65.23	75.93	**70.43**	**78.75**
h-d2v (I)	✓/ ✓	58.59	69.79	66.99	75.63
h-d2v (I+O)	✓/ ✓	**66.64**	75.19	68.96	76.61

Table 5: F_1 on DBLP when newcomers are discarded.

5.1 Datasets and Experimental Settings

We use three datasets from the academic paper domain, *i.e.,* NIPS[4], ACL anthology[5] and DBLP[6], as shown in Table 3. They all contain full text of papers, and are of small, medium, and large size, respectively. We apply ParsCit[7] (Councill et al., 2008) to parse the citations and bibliography sections. Each identified citation string referring to a paper in the same dataset, *e.g.,* [1] or (Author et al., 2018), is replaced by a global paper id. Consecutive citations like [1, 2] are regarded as multiple ground truths occupying one position. Following He et al. (2010), we take 50 words before and after a citation as the citation context.

Gensim (Řehůřek and Sojka, 2010) is used to implement all `w2v` and `d2v` baselines as well as `h-d2v`. We use `cbow` for `w2v` and `pv-dbow` for `d2v`, unless otherwise noted. For all three baselines, we set the (half) context window length to 50. For `w2v`, `d2v`, and the `pv-dm`-based initialization of `h-d2v`, we run 5 epochs following Gensim's default setting. For `h-d2v`, its iteration is set to 100 epochs with 1000 negative samples. The dimension size k of all approaches is 100. All other parameters in Gensim are kept as default.

5.2 Document Classification

In this task, we classify the research fields of papers given their vectors learned on DBLP. To obtain labels, we use Cora[8], a small dataset of Computer Science papers and their field categories. We keep the first levels of the original categories, *e.g.,* "Artificial Intelligence" of "Artificial Intelligence - Natural Language Processing", leading to 10 unique classes. We then intersect the dataset with DBLP, and obtain 5,975 labeled papers.

For `w2v` and `h-d2v` outputing both IN and OUT document vectors, we use IN vectors or concatenations of both vectors as features. For newcomer papers without `w2v` vectors, we use zero vectors instead. To enrich the features with network structure information, we also try concatenating them with the output of DeepWalk (Perozzi et al., 2014), a representative network embedding model. The model is trained on the citation network of DBLP with an existing implementation[9] and default parameters. An SVM classifier with RBF kernel is used. We perform 5-fold cross validation, and report Macro- and Micro-F_1 scores.

5.2.1 Classification Performance

In Table 4, we demonstrate the classification results. We have the following observations.

First, adding DeepWalk information almost always leads to better classification performance, except for Macro-F_1 of the `d2v-cac` approach.

Second, owning to different context awareness, `d2v-cac` consistently outperforms `d2v-nc` in terms of all metrics and settings.

Third, `w2v` has the worst performance. The reason may be that `w2v` is neither content aware nor newcomer friendly. We will elaborate more on the impacts of the two properties in Section 5.2.2.

Finally, no matter whether DeepWalk vectors are used, `h-d2v` achieves the best F_1 scores. However, when OUT vectors are involved, `h-d2v` with DeepWalk has slightly worse performance. A possible explanation is that, when `h-d2v` IN and DeepWalk vectors have enough information to train the SVM classifiers, adding another 100 features (OUT vectors) only increase the parameter

[4] https://cs.nyu.edu/ roweis/data.html

[5] http://clair.eecs.umich.edu/aan/index.php (2013 release)

[6] http://zhou142.myweb.cs.uwindsor.ca/academicpaper.html This page has been unavailable recently. They provide a larger CiteSeer dataset and a collection of DBLP paper ids. To better interpret results from the Computer Science perspective, we intersect them and obtain the DBLP dataset.

[7] https://github.com/knmnyn/ParsCit

[8] http://people.cs.umass.edu/~mccallum/data.html

[9] https://github.com/phanein/deepwalk

Model	NIPS				ACL Anthology				DBLP			
	Rec	MAP	MRR	nDCG	Rec	MAP	MRR	nDCG	Rec	MAP	MRR	nDCG
w2v (cbow, I4I)	5.06	1.29	1.29	2.07	12.28	5.35	5.35	6.96	3.01	1.00	1.00	1.44
w2v (cbow, I4O)	12.92	**6.97**	**6.97**	8.34	15.68	8.54	8.55	10.23	13.26	7.29	7.33	8.58
d2v-nc (pv-dbow, cosine)	14.04	3.39	3.39	5.82	21.09	9.65	9.67	12.29	7.66	3.25	3.25	4.23
d2v-cac (same as d2v-nc)	14.61	4.94	4.94	7.14	28.01	11.82	11.84	15.59	15.67	7.34	7.36	9.16
NPM (Huang et al., 2015b)	7.87	2.73	3.13	4.03	12.86	5.98	5.98	7.59	6.87	3.28	3.28	4.07
h-d2v (random init, I4O)	3.93	0.78	0.78	1.49	30.98	16.76	16.77	20.12	17.22	8.82	8.87	10.65
h-d2v (pv-dm retrofitting, I4O)	**15.73**	6.68	6.68	**8.80**	**31.93**	**17.33**	**17.34**	**20.76**	**21.32**	**10.83**	**10.88**	**13.14**

Table 6: Top-10 citation recommendation results (dimension size $k = 100$).

space of the classifiers and the training variance. For w2v with or without DeepWalk, it is also the case. This may be because information in w2v's IN and OUT vectors is fairly redundant.

5.2.2 Impacts of Content Awareness and Newcomer Friendliness

Because content awareness and newcomer friendliness are highly correlated in Table 1, to isolate and study their impacts, we decouple them as follows. In the 5,975 labeled papers, we keep 2,052 with at least one citation, and redo experiments in Table 4. By carrying out such controlled experiments, we expect to remove the impact of newcomers, and compare all approaches only with respect to different content awareness. In Table 5, we provide the new scores obtained.

By comparing Tables 4 and 5, we observe that w2v benefits from removing newcomers with zero vectors, while all newcomer friendly approaches get lower scores because of fewer training examples. Even though the change, w2v still cannot outperform the other approaches, which reflects the positive impact of content awareness on the classification task. It is also interesting that DeepWalk becomes very competitive. This implies that structure-based methods favor networks with better connectivity. Finally, we note that Table 5 is based on controlled experiments with intentionally skewed data. The results are not intended for comparison among approaches in practical scenarios.

5.3 Citation Recommendation

When writing papers, it is desirable to recommend proper citations for a given context. This could be achieved by comparing the vectors of the context and previous papers. We use all three datasets for this task. Embeddings are trained on papers before 1998, 2012, and 2009, respectively. The remaining papers in each dataset are used for testing.

We compare h-d2v with all approaches in Section 4.2, as well as NPM[10] (Huang et al., 2015b) mentioned in Section 2, the first embedding-based approach for the citation recommendation task. Note that the inference stage involves interactions between word and document vectors and is nontrivial. We describe our choices as below.

First, for w2v vectors, Nalisnick et al. (2016) suggest that the IN-IN similarity favors word pairs with similar functions (*e.g.,* "red" and "blue"), while the IN-OUT similarity characterizes word co-occurrence or compatibility (*e.g.,* "red" and "bull"). For citation recommendation that relies on the compatibility between context words and cited papers, we hypothesize that the IN-for-OUT (or I4O for short) approach will achieve better results. Therefore, for w2v-based approaches, we average IN vectors of context words, then score and and rank OUT document vectors by dot product.

Second, for d2v-based approaches, we use the learned model to infer a document vector $\mathbf{d}$ for the context words, and use $\mathbf{d}$ to rank IN document vectors by cosine similarity. Among multiple attempts, we find this choice to be optimal.

Third, for h-d2v, we adopt the same scoring and ranking configurations as for w2v.

Finally, for NPM, we adopt the same ranking strategy as in Huang et al. (2015b). Following them, we focus on top-10 results and report the Recall, MAP, MRR, and nDCG scores.

5.3.1 Recommendation Performance

In Table 6, we report the citation recommendation results. Our observations are as follows.

First, among all datasets, all methods perform relatively well on the medium-sized ACL dataset. This is because the smallest NIPS dataset provides

[10] Note that the authors used $n = 1000$ for negative sampling, and did not report the number of training epoches. After many trials, we find that setting the number of both the negative samples and epoches at 100 to be relatively effective and affordable w.r.t. training time.

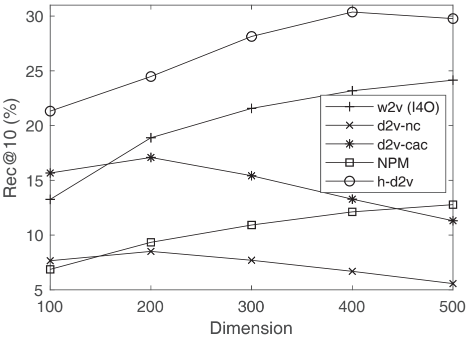


Figure 3: Varying k on DBLP. The scores of w2v keeps increasing to 26.63 at $k = 1000$, and then begins to drop. Although at the cost of a larger model and longer training/inference time, it still cannot outperform h-d2v of 30.37 at $k = 400$.

too few citation contexts to train a good model. Moreover, DBLP requires a larger dimension size k to store more information in the embedding vectors. We increase k and report the Rec@10 scores in Figure 3. We see that all approaches have better performance when k increases to 200, though d2v-based ones start to drop beyond this point.

Second, the I4I variant of w2v has the worst performance among all approaches. This observation validates our hypothesis in Section 5.3.

Third, the d2v-cac approach outperforms its variant d2v-nc in terms of all datasets and metrics. This indicates that context awareness matters in the citation recommendation task.

Fourth, the performance of NPM is sandwiched between those of w2v's two variants. We have tried our best to reproduce it. Our explanation is that NPM is citation-as-word-based, and only depends on citation contexts for training. Therefore, it is only context aware but neither content aware nor newcomer friendly, and behaves like w2v.

Finally, when retrofitting pv-dm, h-d2v generally has the best performance. When we substitute pv-dm with random initialization, the performance is deteriorated by varying degrees on different datasets. This implies that content awareness is also important, if not so important than context awareness, on the citation recommendation task.

5.3.2 Impact of Newcomer Friendliness

Table 7 analyzes the impact of newcomer friendliness. Opposite from what is done in Section 5.2.2, we only evaluate on testing examples where at least a ground-truth paper is a newcomer. Please note that newcomer unfriendly approaches do not necessarily get zero scores. The table shows that newcomer friendly approaches are superior to unfriendly ones. Note that, like Table 5, this table is also based on controlled experiments and not intended for comparing approaches.

Model	Newcomer Friendly	Rec	MAP	MRR	nDCG
w2v (I4O)	×	3.64	3.23	3.41	2.73
NPM	×	1.37	1.13	1.15	0.92
d2v-nc	✓	6.48	3.52	3.54	3.96
d2v-cac	✓	**8.16**	**5.13**	**5.24**	**5.21**
h-d2v	✓	6.41	4.95	5.21	4.49

Table 7: DBLP results evaluated on 63,342 citation contexts with newcomer ground-truth.

Category	Description
Weak	Weakness of cited approach
CoCoGM	Contrast/Comparison in Goals/Methods (neutral)
CoCo-	Work stated to be superior to cited work
CoCoR0	Contrast/Comparison in Results (neutral)
CoCoXY	Contrast between 2 cited methods
PBas	Author uses cited work as basis or starting point
PUse	Author uses tools/algorithms/data/definitions
PModi	Author adapts or modifies tools/algorithms/data
PMot	This citation is positive about approach used or problem addressed (used to motivate work in current paper)
PSim	Author's work and cited work are similar
PSup	Author's work and cited work are compatible/provide support for each other
Neut	Neutral description of cited work, or not enough textual evidence for above categories, or unlisted citation function

Table 8: Annotation scheme of citation functions in Teufel et al. (2006).

5.3.3 Impact of Context Intent Awareness

In this section, we analyze the impact of context intent awareness. We use Teufel et al. (2006)'s 2,824 citation contexts[11] with annotated citation functions, *e.g.,* emphasizing weakness (Weak) or using tools/algorithms (PBas) of the cited papers. Table 8 from Teufel et al. (2006) describes the full annotating scheme. Teufel et al. (2006) also use manual features to evaluate citation function classification. To test all models on capturing context intents, we average all context words' IN vectors (trained on DBLP) as features. Noticing that pv-dbow does not output IN word vectors, and OUT vectors do not provide reasonable results, we use pv-dm here instead. We use SVM with RBF

[11] The number is 2,829 in the original paper. The inconsistency may be due to different regular expressions we used.

Query and Ground Truth	Result Ranking of `w2v`	Result Ranking of `d2v-cac`	Result Ranking of `h-d2v`
. . . We also evaluate our model by computing the machine translation BLEU score (Papineni et al., 2002) using the Moses system (Koehn et al., 2007). . . (Papineni et al., 2002) **BLEU: a Method for Automatic Evaluation of Machine Translation** (Koehn et al., 2007) **Moses: Open Source Toolkit for Statistical Machine Translation**	1. HMM-Based Word Alignment in Statistical Translation 2. Indirect-HMM-based Hypothesis Alignment for Combining Outputs from Machine Translation Systems 3. The Alignment Template Approach to Statistical Machine Translation . . . 9. **Moses: Open Source Toolkit for Statistical Machine Translation** 57. **BLEU: a Method for Automatic Evaluation of Machine Translation**	1. Discriminative Reranking for Machine Translation 2. Learning Phrase-Based Head Transduction Models for Translation of Spoken Utterances 3. Cognates Can Improve Statistical Translation Models . . . 6. **BLEU: a Method for Automatic Evaluation of Machine Translation** 29. **Moses: Open Source Toolkit for Statistical Machine Translation**	1. **BLEU: a Method for Automatic Evaluation of Machine Translation** 2. Statistical Phrase-Based Translation 3. Improved Statistical Alignment Models 4. HMM-Based Word Alignment in Statistical Translation 5. **Moses: Open Source Toolkit for Statistical Machine Translation**

Table 9: Papers recommended by different approaches for a citation context in Zhao and Gildea (2010).

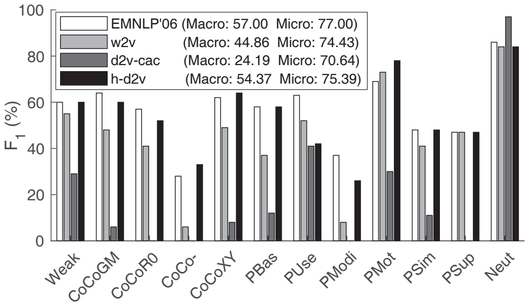


Figure 4: F$_1$ of citation function classification.

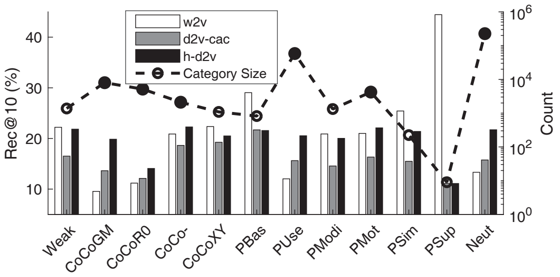


Figure 5: Rec@10 w.r.t. citation functions.

kernels and default parameters. Following Teufel et al. (2006), we use 10-fold cross validation.

Figure 4 depicts the F$_1$ scores. Scores of Teufel et al. (2006)'s approach are from the original paper. We omit `d2v-nc` because it is very inferior to `d2v-cac`. We have the following observations.

First, Teufel et al. (2006)'s feature-engineering-based approach has the best performance. Note that we cannot obtain their original cross validation split, so the comparison may not be fair and is only for consideration in terms of numbers.

Second, among all embedding-based methods, `h-d2v` has the best citation function classification results, which is close to Teufel et al. (2006)'s.

Finally, the `d2v-cac` vectors are only good at Neutral, the largest class. On the other classes and global F$_1$, they are outperformed by `w2v` vectors.

To study how citation function affects citation recommendation, we combine the 2,824 labeled citation contexts and another 1,075 labeled contexts the authors published later to train an SVM, and apply it to the DBLP testing set to get citation functions. We evaluate citation recommendation performance of `w2v` (I4O), `d2v-cac`, and `h-d2v` on a per-citation-function basis. In Figure 5, we break down Rec@10 scores on citation functions. On the six largest classes (marked by solid dots), `h-d2v` outperforms all competitors.

To better investigate the impact of context intent awareness, Table 9 shows recommended papers of the running example of this paper. Here, Zhao and Gildea (2010) cited the BLEU metric (Papineni et al., 2002) and Moses tools (Koehn et al., 2007) of machine translation. However, the additional words "machine translation" lead both `w2v` and `d2v-cac` to recommend many machine translation papers. Only our `h-d2v` manages to recognize the citation function "using tools/algorithms (PBas)", and concentrates on the citation intent to return the right papers in top-5 results.

6 Conclusion

We focus on the hyper-doc embedding problem. We propose that hyper-doc embedding algorithms should be content aware, context aware, newcomer friendly, and context intent aware. To meet all four criteria, we propose a general approach, `hyperdoc2vec`, which assigns two vectors to each hyper-doc and models citations in a straightforward manner. In doing so, the learned embeddings satisfy all criteria, which no existing model is able to. For evaluation, paper classification and citation recommendation are conducted on three academic paper datasets. Results confirm the effectiveness of our approach. Further analyses also demonstrate that possessing the four properties helps `h-d2v` outperform other models.

References

Matthew Berger, Katherine McDonough, and Lee M. Seversky. 2017. cite2vec: Citation-driven document exploration via word embeddings. *IEEE Trans. Vis. Comput. Graph.* 23(1):691–700.

David A. Cohn and Thomas Hofmann. 2000. The missing link - A probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000*. pages 430–436.

Isaac G. Councill, C. Lee Giles, and Min-Yen Kan. 2008. Parscit: an open-source CRF reference string parsing package. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pages 708–716.

Travis Ebesu and Yi Fang. 2017. Neural citation network for context-aware citation recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pages 1093–1096.

Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. 2016. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*. pages 260–269.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1606–1615.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*. pages 1606–1611.

Soumyajit Ganguly and Vikram Pudi. 2017. Paper2vec: Combining graph and text information for scientific paper representation. In *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017*. pages 383–395.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pages 855–864.

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and C. Lee Giles. 2010. Context-aware citation recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*. pages 421–430.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Volume 2: Short Papers*. pages 30–34.

Hongzhao Huang, Larry P. Heck, and Heng Ji. 2015a. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *CoRR* abs/1504.07678.

Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C. Lee Giles, and Lior Rokach. 2012. Recommending citations: translating papers into references. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12*. pages 1910–1914.

Wenyi Huang, Zhaohui Wu, Liang Chen, Prasenjit Mitra, and C. Lee Giles. 2015b. A neural probabilistic model for context based citation recommendation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. pages 2404–2410.

Saurabh Kataria, Prasenjit Mitra, and Sumit Bhatia. 2010. Utilizing context in generative bayesian models for linked corpus. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*. pages 1188–1196.

Qing Lu and Lise Getoor. 2003. Link-based classification. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*. pages 496–503.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013.*. pages 3111–3119.

Eric T. Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. 2016. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Companion Volume*. pages 83–84.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. .

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. pages 311–318.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*. pages 701–710.

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*. ELRA, Valletta, Malta, pages 45–50. http://is.muni.cz/publication/884893/en.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.* 27(2):443–460.

Kazunari Sugiyama and Min-Yen Kan. 2010. Scholarly paper recommendation via user's recent research interests. In *Proceedings of the 2010 Joint International Conference on Digital Libraries, JCDL 2010*. pages 29–38.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*. pages 1333–1339.

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015a. PTE: predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pages 1165–1174.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015b. LINE: large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*. pages 1067–1077.

Jie Tang and Jing Zhang. 2009. A discriminative approach to topic-based citation recommendation. In *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009*. pages 572–579.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *EMNLP 2007, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. pages 103–110.

Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. 2016. Max-margin deepwalk: Discriminative learning of network representation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*. pages 3889–3895.

Suhang Wang, Jiliang Tang, Charu C. Aggarwal, and Huan Liu. 2016. Linked document embedding for classification. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*. pages 115–124.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*. pages 250–259.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network representation learning with rich text information. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*. pages 2111–2117.

Shaojun Zhao and Daniel Gildea. 2010. A fast fertility hidden markov model for word alignment using MCMC. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010*. pages 596–605.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016*. pages 425–434.