A Study of Distributed Representations for Figures of Research Articles

Saar Kuzi(✉) and ChengXiang Zhai

University of Illinois at Urbana-Champaign, Urbana, Illinois, USA
{skuzi2,czhai}@illinois.edu

Abstract. Figures of research articles are entities that can be directly used in many application systems to assist researchers, making the representation of figures a problem worth studying. In this paper, we study the effectiveness of distributed representations, learned using deep neural networks, for figures. We learn representations using both text and image data and compare different model architectures and loss functions for the task. Furthermore, to overcome the lack of training data for the task, we propose and study a novel weak supervision approach for learning embedding vectors and show that it is more effective than using some of the pre-trained neural models as suggested by recent works. Experimental results using figures from the ACL Anthology show that distributed representations for research figures can be more effective than the previously studied bag-of-words representations. Yet, combining the two approaches can further improve performance. Finally, the results also show that these representations, while effective in general, can be sensitive to the learning approach used and that using both image data and text and a simple model architecture is the most effective approach.

1 Introduction

Figures are entities in research articles that play an essential role in scientific communications. To accelerate research, it is important to develop tools to assist researchers in accessing and digesting figures. Figure representation is a fundamental problem in all applications involving figures. Different from general images, figures are complex research entities that are associated with various sources of data of various modalities, posing unique novel challenges for representation learning. Thus, the study of how to optimize representation specifically for research figures is crucial. Despite that, this problem has not been well studied in previous works. The dominant approach explored in the existing studies is to represent a figure by its companion text data in an article using the bag-of-words approach [15]. Using this representation of figures has some limitations. First, it does not consider any other types of non-textual features, such as image features. Second, it has limited capability in accommodating the inexact matching of semantically related words.

To address the limitations of the previous work, we study a new view of representation for figures, namely deep neural network-based distributed representations. Learning distributed representations for many real-world entities has

D. Hiemstra et al. (Eds.): ECIR 2021, LNCS 12656, pp. 284–297, 2021.
https://doi.org/10.1007/978-3-030-72113-8_19

been shown to be very successful in recent years [19,25]. The main idea behind the different approaches in this scope is to learn embeddings of those entities using large data sets where the goal of learning is to capture the complex relations between the entities. For example, learning an embedding representation of words [6,19] has proven to be useful for many text applications. Specifically, word embeddings can effectively address some of the limitations of bag-of-words representations, such as measuring the semantic similarity between words.

In this paper, our goal is to study the effectiveness of distributed representations for figures, exploring the learning of such a representation from multiple views. Specifically, we focus on using both image data and text for learning representations with different model architectures and loss functions to understand how sensitive the embeddings are to the learning approach and the features used.

One technical challenge in learning deep neural network-based representations is that it requires massive amounts of data that is not available for this domain. While word embeddings can be easily learned by leveraging the co-occurrences of words in large amounts of text data, the amount of figure data is quite limited. To overcome this problem, we propose and study two strategies. The first is to leverage massively pre-trained models on general data (e.g., BERT [6]). The second is a novel weak supervision approach that can generate a large amount of training data by leveraging the already existing citation relations between research articles.

We used a collection of figures from the ACL Anthology to empirically study the effectiveness of different representations by their ability to measure the semantic similarity between research figures. We also study the effectiveness of embeddings in the downstream application of recommending figures of interest based on an input (query) figure. The results show that embeddings are generally more effective than bag-of-words, yet combining them is the best performing approach. Another finding is that the pre-trained image/text embeddings have limited effectiveness compared to the weak supervision approach and even the bag-of-words approach. Finally, the results show that the effectiveness of embeddings for figures can be somewhat sensitive to the learning technique. Specifically, the relatively simple model architectures are the most effective ones, text features are more effective than image features, and combining image and text features is the most successful approach.

2 Related Work

There has been growing interest recently in learning vector representations of real-world entities using deep neural networks. This led to the development and study of various embedding models for representing different entities such as words [6,19], sentences [17], and images [22]. Our work can be regarded as the first one to study the effectiveness of embedding-based representations for figures.

Learning embeddings using neural networks often requires massive amounts of data. To address this, there has been an active research direction exploring the use of weak supervision for learning [3,5]. Our work adds to the existing work a new line of application of weak supervision for learning figure embeddings.

There have been several previous works that studied various figure retrieval and mining tasks [2,10,13,16,18]. These previous works mostly relied on the bag-of-words representation of figures. In this work, we explore distributed representations of figures that can benefit a variety of tasks that involve figures.

Previous works have studied the joint embedding of images and text, focusing mostly on images that contain different objects and text that identifies the objects and the interactions between them (e.g., "An apple on a table") [7,8,14,21,23,25]. The main idea in many of these works was to embed image and text to the same space. Learning joint embeddings for image and text aims to find a common representation that can explain both and is thus less appropriate for research figures in which image and text are often two types of complementary information. Thus, in this paper, we learn text- and image-based features separately and combine them using a third model. Using this strategy is sufficient for studying the different aspects of the problem that we are interested in, such as the effectiveness of various architectures for image/text modeling, the effectiveness of image and text feature combination, and the effectiveness of pretraining vs. weak supervision. We thus leave the study on finding the optimal integration of image and text features for figures for future work.

3 Figure Embeddings

Problem Definition: A collection of figures F_D can be generated using a collection of research articles D by extracting the figures from all articles. Each figure can be associated with different types of data of different modalities. For example, a figure can be associated with a caption, the abstract section of its article, an image, and a set of numbers. In this study, as a first step, we focus on learning figure embeddings using only text and image data. Given two figures in the collection, f_i and f_j, the goal is to learn corresponding vectors in a continuous space, f_i and f_j, such that the distance between them in that space is inversely proportional to their semantic similarity. In this paper, we use neural networks to learn these representations of figures.

Textual Representation of Figures: While the image data of a figure is well defined, the textual data for a figure is not readily available. In the general case, the article that contains the figure can be used to extract text that directly describes it (e.g., the figure caption) and text that does not directly describe it, but is related to its topic (e.g., parts of the abstract section). One previous work [15] has explored the effectiveness of using different types of textual data for figure representation to be used for the figure retrieval task. Based on the findings of that work, we generate a textual representation for a figure as follows. We use the caption of the figure, concatenated together with the text in the article that directly describes the figure, for the figure representation. To extract this text, first, the locations in the article where the figure is mentioned are identified. Then, the sentence that directly mentions the figure, one sentence before it, and one sentence after it, are extracted. (In the case of several mentions for the figure, all the text which was extracted is merged.) We use this text as a single textual

input which resulted in a good enough performance. In future work, we plan to take into account the sources of those different texts in the learning approach.

Model Architecture: To learn figure embeddings using neural networks, we use the Siamese architecture [4]. According to this architecture, given two figures, the same model is used to generate embeddings for both of them. Then, the dot product between the figure vectors is used as a semantic similarity score. The Siamese model is appropriate for our scenario since the two figures are entities of the same type and we also assume the relationship between them is symmetric. We note that the symmetry assumption may not always hold but is still useful to learn meaningful representations; we thus leave the treatment of asymmetric relationships for future work. The model for our figure embedding approach is composed of three sub-models: (1) An image model that generates visual features. (2) A text model that generates textual features. (3) A fusion model that combines the image and text features. While the image and the text model are both Siamese models, the fusion model is a feed-forward neural network model.

Text Models: To generate textual features, we experimented with three models to explore varying levels of complexity, compare auto-regressive to non-auto-regressive models, and compare pre-training to weak supervision-based training. The first model we used is LSTM [11] that generates features for a text using a recurrent neural network. Specifically, our LSTM-based model contains a word embedding layer (learned from scratch) which is followed by a single LSTM layer, where the weights of the last hidden state of the LSTM layer are used as the textual features. The second model we use is Bi-LSTM [9]. This model is similar to LSTM but has a higher level of complexity since it models dependencies using both directions of the text. As in the case of the LSTM model, we use a word embedding layer which is followed by the Bi-LSTM layer. Additionally, the Bi-LSTM layer generates two sets of features (backward and forward). The two sets of features are concatenated, a dropout layer is added on top of this concatenation, and a final dense layer is added to obtain the textual features. The last model we use is BERT [6] which uses transformers and self-attention mechanisms to learn dependencies in text. This model was shown to achieve state-of-the-art performance in many NLP tasks, where the main approach that was taken is to pre-train the model using a very large amount of text data and then fine-tune the output of the model for the specific task. We experiment with three versions of this model. In the first one, we use a pre-trained model and treat the pooled output as the textual features. In the second version, we add a dropout layer, a dense layer with a Relu activation, and a final dense layer on top of the pooled output. Then, we learn the weights of those dense layers using the Siamese architecture; the output of the final dense layer serves as the textual features. In the third version, we use the same model as in the second one but also fine-tune the last layer of BERT.

Image Models: Previous works on using neural networks for computer vision leveraged massive amounts of data which enabled the learning of complex models

with remarkable performance. Another technique that is highly effective for computer vision is transfer learning in which a model is trained using large amounts of data and then is fine-tuned for a specific task. In this work, our goal is to generate effective image features for figures. This is challenging, however, since we do not have available massive amounts of image training data. Furthermore, since images of figures are quite different than images in the massive training data sets (e.g., ImageNet), it is not clear how pre-training will be useful for our scenario. To better understand these issues, we experiment with two models as follows. The first model that we use is a simple Convolutional Neural Network Model (CNN) which is fully trained using the figure image data. The model is composed of two convolutional layers, a max-pooling layer, a dropout layer, a dense layer with Relu activation, and a final dense layer. The second model we use is DenseNet [12] which uses densely connected convolutional networks. This model has higher complexity than the simple CNN and we use it since it was previously shown to be very effective for image representation. We use three versions of this model. In the first one, we use a pre-trained model with ImageNet to generate image features (no fine-tuning). In the second version, we add layers on top of the DenseNet model including a dropout layer, a dense layer with Relu activation, and a final dense layer. We then learn the parameters of the dense layers using the Siamese model. In the third one, we use the same architecture as in the second version but additionally fine-tune the last dense block of the DenseNet model.

Fusion Model: To combine the image and text features, we concatenate them and use a batch normalization layer on top of that. Finally, we use a single dense layer to generate the figure embedding. We take this approach since we are interested in obtaining a single embedding vector for a figure using different types of features.

Loss Function: We assume that each pair of figures, f_i and f_j, is associated with a numeric semantic similarity score $R_{i,j} \in \mathbb{R}$ (larger values of $R_{i,j}$ correspond to greater similarity). A semantic similarity label $L_{i,j} \in \{0, 1\}$ can be generated using $R_{i,j}$ by setting $L_{i,j}$ to 1 when $R_{i,j} > 0$ and setting $L_{i,j}$ to 0 otherwise. We experiment with three loss functions. The first one is the Cross-Entropy loss, computed using the Sigmoid of the dot product between the two vectors and the semantic similarity label, $CE(f_i \cdot f_j, L_{i,j})$. Secondly, we use the Mean Squared Error loss, computed using the dot product between the two vectors and the semantic similarity score, $MSE(f_i \cdot f_j, R_{i,j})$. Finally, we use the triplet hinge loss [22]. The triplet hinge loss is defined for a triplet of figures, comprised of a query figure f_q, a positive figure f_+ (i.e., a related figure), and a negative figure (f_-). This loss is defined as: $max(0, 1 + f_q \cdot f_- - f_q \cdot f_+)$. The main idea is that we want a figure to be closer to a related figure compared to an unrelated figure.

4 Weak Supervision for Figure Embeddings

Since we are dealing with a novel problem in this paper, an important issue that needs to be addressed is how to collect training data. Furthermore, since we are interested in using deep neural networks, there is a need for a large set of training examples. To address this challenge, since log data was not available to us, we propose a novel approach for collecting data for the task using weak supervision.[1] This approach allows us to leverage large amounts of training data that already exist. Specifically, to generate training data, we leverage existing relations between research articles. First, since we know that two articles are related if one is cited by the other, we assume that two figures are semantically similar if they appear in two articles with a citation relation. Secondly, we assume that any two figures that are in the same article are also semantically similar. Although both kinds of relations may be noisy, we expect that most relations are meaningful semantic associations and the learned embedding vectors to be meaningful as in the case of word embeddings where there are also noisy word associations, but they do not significantly affect the results. Comparing the two types of relations, it is reasonable to assume that two figures that appear in the same article are more likely to be more semantically similar than two figures that appear in citing articles and that the latter should be more similar than a random pair of figures. Based on this intuition, we set the semantic similarity score of two figures in citing articles to be lower than the score of two figures in the same paper. Finally, to generate negative examples, we randomly sample pairs of figures from the collection. Formally, given two figures f_i and f_j, extracted form the articles $d(f_i)$ and $d(f_j)$, respectively, and given that $C(d(f_i))$ is the set of articles that cite $d(f_i)$ or are cited by it, the semantic similarity score $R_{i,j}$ is set to 1 if $d(f_i) = d(f_j)$, 0.6 if $d(f_j) \in C(d(f_i))$, and 0 otherwise.

When using this data for training the image model, some modifications need to be made. This is the case since semantically similar figures, as defined by our approach, may have images that are not visually similar. Our goal for the image model is to be able to generate features that can help measure the visual similarity between figures. For this reason, we filter out pairs of figures which are not visually similar enough (we use the Structural Similarity Index (SSIM) [24] with a threshold of 0.5 for filtering out pairs, and a threshold of 0.3 for sampling negative pairs). Finally, we do not make a differentiation between figures in the same paper and figures in citing papers since these relationships may not be indicative of different levels of visual similarity. Taking this approach, a pair of figures will be assigned only with a binary relevance label in the case of the image model (and consequently we do not use the MSE loss).

[1] While there are publicly available large collections of figures [20], they do not provide any relations between figures for the purpose of representation learning.

5 Empirical Study

5.1 Experimental Setup

Collection of Figures: A collection of figures was built using the ACL Anthology (aclweb.org/anthology). 40,367 articles whose copyright belongs to ACL and were published until October 2018 were crawled. Using those articles, a collection of 84,340 figures was created. The PdfFigures toolkit (pdffigures2.allenai.org) was used to extract the figure images. The Grobid toolkit was used to extract the full text from the PDF files of the articles (github.com/kermitt2/grobid).

Data Pre-processing: Text data was Porter stemmed and stopwords were removed (using the INQUERY list). Figures with an associated text, after pre-processing, of less than 5 words were removed. Images of figures were resized to fit a $224 \times 224 \times 3$ matrix and were normalized by a factor of 255.

Training Data: 947,335 pairs of figures in citing articles and 202,944 pairs of figures in the same article were used as related figures. After adding random pairs as negative examples, we ended up having about 2M pairs of figures for training the text network. For the image network, after filtering out images that were not visually similar enough, we ended up with about 300K figure pairs for training. For training the fusion network, since we are interested in figures with both text and image data, we used about 1M pairs after filtering out figures with no image data. In this work, we train all three components of the model (i.e., the image model, text model, and fusion model) separately, due to our limited data. For the evaluation of the different approaches, we only use figures for which both image and text data is available to make it as realistic as possible (57K figures).

Neural Network Implementation: The neural network was implemented using the TensorFlow library. We set the values of the different parameters based on findings in recent works in the text and image domain. All models were trained for 3 epochs using the Adam optimizer with a batch size of 64 and a learning rate of 0.01. The vocabulary size was set to the 1000 most frequent words in the training data. We used only the first 100 words in the text data of a figure (the figure caption was concatenated first) due to BERT's limitation on the input size and the limited effectiveness of LSTM for long sequences. The embedding size was set to 50 for all models which means that the number of hidden layers in LSTM/Bi-LSTM was set to 50 as well as the size of the final dense layer in the other models (our preliminary experiments showed that a larger size of 100 is less effective). The size of the dense layer on top of BERT, DenseNet, and CNN was set to 100. The dropout rate was set to 0.5. The word embedding layer dimension for the LSTM/Bi-LSTM model was set to 100. For BERT, we used a model with 12 layers, 768 hidden units, and 12 attention heads. For DenseNet, we used a 121-layer model. For the CNN model, we used convolutional layers with 32 filters and a kernel size of 3×3.

Baselines: Since one of the main research questions that we study is whether embeddings can improve over the currently used bag-of-words representations,

we compare our model with two representative baseline methods: tf.idf and LDA. For the LDA baseline [1], we learn a model with 50 topics and use the figure distribution over topics as its representation. The vocabulary used for both models was also restricted to 1000 frequent words.

5.2 Experimental Results

Semantic Similarity Prediction: To evaluate the effectiveness of the different representations in measuring the semantic similarity between figures, a binary classification task was performed. Given two figure vectors, the cosine function was used to get a similarity score which was then transformed into a binary label using a threshold. Since the threshold value can vary depending on the representation type, a validation set was used to set it (selected from $\{0.1, 0.2, ..., 0.9\}$). Three test sets were created for the evaluation. In the first one, denoted "Same", we used 500 pairs of figures that appear in the same article (related figures) and 500 randomly sampled (unrelated) pairs. In the second one, denoted "Citing", we used 500 pairs of figures that appear in citing articles (related figures) and 500 unrelated pairs. Finally, in the third set, denoted "Accuracy", the first two sets were combined. (All selected pairs were removed from the training set.) The sets were balanced such that the accuracy of a random baseline is 0.5. The results are presented in Table 1 for using text and image features separately and in Table 3 for the fusion model. For pre-trained models that were fine-tuned (i.e., BERT and DenseNet), we added the term "(tuned)" when only the dense layers on top of the model were fine-tuned and "(tuned+)" when the dense layers and also part of the model were fine-tuned.

According to the results in Table 1, most text-based and image-based representations perform better than a random baseline. Focusing on the embedding models which use text features only, we can see that for the LSTM/Bi-LSTM model the best performance is achieved for the MSE loss, while for the tuned BERT models there are no large differences between the different loss functions. Overall, based on the results, the best text-based embedding model is LSTM. A possible reason for this might be its relatively small number of parameters and the size of the training data set. Also, it is interesting to see that it outperforms the pre-trained BERT model which might be attributed to the unique vocabulary used in ACL research articles. Comparing the embedding models to the baselines, we can see that LSTM/Bi-LSTM largely outperforms all baselines including tf.idf, LDA, and the pre-trained BERT model. We can also see from the results that tf.idf is the strongest baseline. For this reason, we focus on comparing our embedding approaches mostly to this baseline in the remainder of the evaluation section. Focusing on BERT, we can see that fine-tuning can improve its performance, but resulting in overall effectiveness that is still low. Another finding from the table is that the improvements of the embedding methods over the bag-of-words baselines for the case of citing figures are much larger than the case of figures in the same article. This might be due to the soft matching nature of distributed (dense) representations and their ability to identify more loosely related figures. Moving on to the image features, we can see that most

Table 1. Semantic similarity prediction: text vs. image.

			Accuracy	Same	Citing
Text features		tf.idf	.720	.818	.622
		LDA	.688	.766	.609
		BERT	.525	.522	.527
	CE	LSTM	.740	.776	.704
		Bi-LSTM	.732	.743	.720
		BERT(tuned)	.533	.535	.530
		BERT(tuned+)	.534	.534	.533
	MSE	LSTM	.802	.831	.772
		Bi-LSTM	.791	.811	.770
		BERT(tuned)	.527	.527	.527
		BERT(tuned+)	.527	.528	.525
	Hinge	LSTM	.505	.508	.501
		Bi-LSTM	.500	.500	.500
		BERT(tuned)	.522	.525	.518
		BERT(tuned+)	.534	.537	.530
Image features		DenseNet	.620	.623	.616
	CE	CNN	.500	.500	.500
		DenseNet(tuned)	.635	.641	.629
		DenseNet(tuned+)	.518	.510	.526
	Hinge	CNN	.662	.663	.661
		DenseNet(tuned)	.630	.655	.605
		DenseNet(tuned+)	.500	.500	.499

of them perform better than a random approach and that the best performing model is CNN. Finally, we can see that using fine-tuning for the DenseNet model can improve its performance. Yet, the performance of the fine-tuned DenseNet model is still not as good as that of CNN. Comparing the image to text features we can see that the text features are more effective.

Table 2. Combining tf.idf with text-based embeddings using an "Oracle".

	Accuracy	Same	Citing
tf.idf	.720	.818	.622
LSTM	.802	.831	.772
BERT(tuned+)	.534	.534	.533
LSTM& tf.idf	.914	.941	.886
BERT(tuned+)& tf.idf	.864	.913	.815

In light of the results in Table 1, an important question that comes up is whether embeddings can replace tf.idf for the textual representation of figures. To answer this, we examine the effectiveness of combining the predictions of tf.idf and embeddings using an oracle in Table 2 which serves as an upper-bound for the performance of such combination. We focus on effective models according to Table 1: LSTM trained with MSE and BERT(tuned+) trained with CE. The results show that this combination is of merit, always outperforming the individual models. Even in the case of BERT, which is not very effective according to Table 1, the combination can improve tf.idf substantially. In this paper, we are mainly interested in studying distributed representations and thus leave the study of how to combine the two approaches for future work.

Table 3. Semantic similarity prediction: fusion model.

		Accuracy	Same	Citing
	tf.idf	.720	.818	.622
	LSTM	.802	.831	.772
	BERT(tuned+)	.534	.534	.533
	CNN	.662	.663	.661
	DenseNet(tuned)	.635	.641	.629
CE	LSTM& CNN	.805	.834	.775
	BERT(tuned+)& CNN	.684	.689	.678
	LSTM& DenseNet(tuned)	.643	.681	.604
	BERT(tuned+)& DenseNet(tuned)	.678	.680	.675
MSE	LSTM& CNN	.838	.866	.809
	BERT(tuned+)& CNN	.699	.704	.693
	LSTM& DenseNet(tuned)	.726	.760	.691
	BERT(tuned+)& DenseNet(tuned)	.693	.698	.687

Next, we analyze the performance of representations that combine both image and text data in Table 3. We focus on studying the combination of the most effective image and text features, based on the results in Table 1. Specifically, we use LSTM trained with MSE, BERT(tuned+) with CE, CNN with Hinge loss, and DenseNet(tuned) with CE. We also focus only on MSE and CE due to the very poor performance of the Hinge loss for the textual features. The results show that for the majority of model combinations, using both features largely outperforms the individual components. This finding supports the idea that image and text features are complementary and represent different aspects of the figure. Finally, we can see that the MSE loss is the best performing for all models and that the best performing model is the LSTM&CNN model.

Figure Recommendation: The goal of this task is to recommend figures to the user that are related to a target figure. To address this problem, a standard

two-phase approach was used. First, using the target figure, an initial retrieval is performed to get an initial figure set. Then, a re-ranking model is used to obtain the recommended figures. To build a test set of target figures for testing, we first collected all figures that have at least 5 more figures in the same article and also 5 figures in citing articles (to result in $p@5 = 1$ at the best scenario). From this set, 500 figures were selected randomly (400 for testing and 100 for validation); all pairs of figures that contained at least one of the target figures were removed from the training set. The performance of the different models is measured using $p@3$ and $p@5$. Since there are no human relevance judgments available for the task, we assume that a figure is relevant if it appears in the same article as the target figure ("Same"), a citing article ("Citing"), or in either (the main performance measure). We note that while this evaluation is not fully realistic, it can still help us make meaningful comparisons between the different approaches. Statistically significant differences between approaches were measured using the two-tailed paired t-test at a 95% confidence level.

Table 4. Retrieval performance of the recommendation task. All differences with tf.idf are statistically significant.

			Same		Citing	
	$p@3$	$p@5$	$p@3$	$p@5$	$p@3$	$p@5$
tf.idf	.298	.228	.354	.276	.057	.048
LSTM	.044	.032	.058	.047	.014	.016
CNN	.000	.001	.001	.003	.001	.002
LSTM& CNN	.051	.039	.066	.054	.015	.014

Table 5. Figure recommendation performance. Statistically significant differences with tf.idf are marked with an asterisk.

			Same		Citing	
	$p@3$	$p@5$	$p@3$	$p@5$	$p@3$	$p@5$
tf.idf	.298	.228	.354	.276	.057	.048
	Cross Entropy (CE)					
LSTM& CNN	.308	.241*	.368	.296*	.060	.056*
BERT(tuned+)& CNN	.296	.227	.352	.277	.056	.050
LSTM& DenseNet(tuned)	.303	.233	.355	.289*	.053	.056*
BERT(tuned+)& DenseNet(tuned)	.303	.229	.357	.279	.054	.050
	Mean Squared Error (MSE)					
LSTM& CNN	.320*	.240*	.380*	.299*	.060	.059*
BERT(tuned+)& CNN	.296	.226	.353	.277	.057	.052
LSTM& DenseNet(tuned)	.313*	.235*	.371*	.287*	.058	.053
BERT(tuned+)& DenseNet(tuned)	.300	.229	.356	.278	.056	.049

First, we study the effectiveness of the retrieval step in Table 4. The performance of three embedding methods (which use text data, image data, and both), trained using the MSE loss, is compared with that of tf.idf. We can see that the tf.idf approach is the most successful. This result is expected since tf.idf relies mostly on exact keyword matching while embedding-based methods rely more on soft matching. Since we are searching over the entire collection, the embedding model may not be discriminative enough.

The performance of the recommendation task is reported in Table 5. To obtain these results, we first perform retrieval using tf.idf and then re-rank the first 100 figures using the cosine similarity between the figure embeddings. The final score for a figure is defined as a linear interpolation between the tf.idf score and the embedding score. The weight for the tf.idf component and the embedding component in the interpolation is determined using a validation set (selected from $\{0.1, 0.2, ..., 0.9\}$; the weights are set to sum up to 1). We experiment with embedding approaches that use both text and image data with the same setting as in Table 3. According to the results in Table 5, we can see that using embeddings on top of the initial retrieval results (tf.idf based) can largely improve the recommendation performance. Specifically, the embedding approaches outperform the baseline in terms of the overall $p@3$ and $p@5$ for the majority of relevant comparisons. Comparing the LSTM model to BERT, we can see that the former is better in the majority of cases. The best embedding model, according to the results, is the LSTM&CNN model with the MSE loss.

Table 6. Figure recommendation example.

LDA graphical representation	
tf.idf	Embeddings
1. The graphical representation of LDA	1. The graphical representation of LDA
2. Graphical models of LDA and DMM	2. Topic model
3. Topic model	3. Graphical representation of strTM
4. Plate notation of our model: MATM	4. Plate notation of our model: MATM
5. LDA plate diagram	5. Graphical representation of (a) BTM, (b) Twitter-BTM

An example target figure with its recommendation list is presented in Table 6. In the table, the caption of the target figure is presented together with the captions of five recommended figures when using either tf.idf or embeddings (LSTM&CNN with MSE); in both cases, tf.idf was used for the initial retrieval. The subject of the figure is the graphical representation of the LDA topic model. Using the tf.idf approach, we get figures that are either equivalent (e.g., "LDA plate diagram"), or diagrams of related models (e.g., "MATM" and "DMM"). When using the embedding approach, we can see that we get more diverse recommendations. This difference can be because using embeddings results in softer matching compared to tf.idf.

6 Conclusions

In this work, we studied the effectiveness of neural network-based figure embeddings. The experimental results showed that figure embeddings outperform the bag-of-words approach in the tasks of semantic similarity prediction and figure recommendation. We also observed that embeddings cannot replace the bag-of-words approach and that combining the two is the best practice. Finally, the results also showed that some learning approaches can be more effective than others. Specifically, using a simple model architecture and combining both image and text features performs the best.

In future work, different methods for combining the different figure features can be studied. Collecting user data to learn more effective representations and improve the evaluation is another possible future direction.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grants No. 1801652 and No. 1937115.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Bockhorst, J.P., Conroy, J.M., Agarwal, S., O'Leary, D.P., Yu, H.: Beyond captions: Linking figures with abstract sentences in biomedical articles. PloS one **7**(7), e39618 (2012)
3. Bordes, A., Weston, J., Usunier, N.: Open question answering with weakly supervised embedding models. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8724, pp. 165–180. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44848-9_11
4. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a" siamese" time delay neural network. In: Advances in Neural Information Processing Systems, pp. 737–744 (1994)
5. Dehghani, M., Zamani, H., Severyn, A., Kamps, J., Croft, W.B.: Neural ranking models with weak supervision. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 65–74. ACM (2017)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Dey, S., Dutta, A., Ghosh, S.K., Valveny, E., Lladós, J., Pal, U.: Learning cross-modal deep embeddings for multi-object image retrieval using text and sketch. arXiv preprint arXiv:1804.10819 (2018)
8. Frome, A., et al.: Devise: a deep visual-semantic embedding model. In: Advances in neural information processing systems, pp. 2121–2129 (2013)
9. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
10. Hearst, M.A., Divoli, A., Guturu, H., Ksikes, A., Nakov, P., Wooldridge, M.A., Ye, J.: Biotext search engine: beyond abstract search. Bioinformatics **23**(16), 2196–2197 (2007)

11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
13. Kim, D., Ramesh, B.P., Yu, H.: Automatic figure classification in bioscience literature. J. Biomed. Inform. **44**(5), 848–858 (2011)
14. Klein, B., Lev, G., Sadeh, G., Wolf, L.: Associating neural word embeddings with deep image representations using fisher vectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4437–4446 (2015)
15. Kuzi, S., Zhai, C.X.: Figure retrieval from collections of research articles. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) ECIR 2019. LNCS, vol. 11437, pp. 696–710. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15712-8_45
16. Kuzi, S., Zhai, C., Tian, Y., Tang, H.: Figexplorer: a system for retrieval and exploration of figures from collections of research articles. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2133–2136 (2020)
17. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
18. Liu, F., Yu, H.: Learning to rank figures within a biomedical article. PloS one **9**(3), e61567 (2014)
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
20. Siegel, N., Lourie, N., Power, R., Ammar, W.: Extracting scientific figures with distantly supervised neural networks. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, pp. 223–232 (2018)
21. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156–3164 (2015)
22. Wang, J., et al.: Learning fine-grained image similarity with deep ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1386–1393 (2014)
23. Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5005–5013 (2016)
24. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
25. Weston, J., Bengio, S., Usunier, N.: Large scale image annotation: learning to rank with joint word-image embeddings. Mach. Learn. **81**(1), 21–35 (2010)