Unsupervised Semantic Generative Adversarial Networks for Expert Retrieval

Shangsong Liang
School of Data and Computer Science, Sun Yat-sen University, China
Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
liangshangsong@gmail.com

ABSTRACT

Sources in computer-based collaborative systems such as webpages can help employees to connect and cooperate with each other. It is natural to enable the systems to look not only for documents but also for experts. In this paper, we study the problem of expert retrieval in enterprise corpora: given a topic, also known as query containing a set of words, identify a rank list of candidate experts who have expertise on the topic. To tackle the problem, we propose an unsupervised semantic two-player minimax game, i.e., our unsupervised semantic generative adversarial networks (USGAN). Unlike almost all the previous generative adversarial networks-based algorithms that require ground truth training data, our USGAN is an unsupervised semantic expert retrieval algorithm that consists of a discriminative network and a generative network aiming at capturing the representations of words and experts in an unsupervised way. Candidates that have similar semantic representations to that of the topic are retrieved as relevant to the topic. Our USGAN would provide inspiration on how to extend the standard GAN and its variants by unsupervised ways to address other retrieval tasks where labelled data are missing. Experimental results on public datasets validate the effectiveness of the proposed expert retrieval algorithm.

CCS CONCEPTS

• **Information systems → Expert search**.

KEYWORDS

Expert Retrieval; Language Models; Generative Adversarial Networks

ACM Reference Format:
Shangsong Liang. 2019. Unsupervised Semantic Generative Adversarial Networks for Expert Retrieval. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3308558.3313625

1 INTRODUCTION

Computer-based collaborative systems have been widely used by many organizations these days. Sources in such systems, such as webpages, database records, agendas, employees' description files and technical reports, can help to connect employees for collaboration and information exchange in the organization. It is natural to enable the systems to look not only for documents, but for entities, such as answers, services, objects, and experts. Our interest is in one type of entity in particular: experts. Accordingly, in this paper, we aim at addressing the *expert retrieval task, also known as expert finding, expert retrieval, or expert search task: given a topic, also called a query containing a set of words, who are the experts with the most expertise on the topic?* Our goal is to retrieve a rank list of experts who have expertise on a specific topic described by an input query, given a set of candidates and a heterogeneous document repository of an organization.

The study of expert retrieval task has gained popularity in research community since the launch of the task at TREC 2005 enterprise track [14]. Algorithms for the task are either supervised or unsupervised. Supervised ones require great manual annotation efforts to obtain training data, resulting in the fact that a limited amount of training data may greatly hinder their applications. In contrast, unsupervised ones, such as those based on language models [8], do not require the training data, and can work well in many cases. In this paper we aim at improving the performance with no training data, and propose **U**nsupervised **S**emantic **G**enerative **A**dversarial **N**etworks (**USGAN** for short), i.e., an unsupervised semantic minimax game, for the expert retrieval task.

GAN (Generative Adversarial Networks [20]) works with a minimax game consisting of a discriminative network D that learns to distinguish whether a given data instance is real or not, and a generative network G that learns to confuse D by generating high quality data. GAN and its variants have been successfully applied into many applications such as those in artificial intelligence, e.g., abstract reasoning [28], clustering [42], and image and sequence generation [27, 57], and have shown their better performance compared to that of many maximum likelihood techniques and traditional deep learning models, e.g., convolutional neural networks [19]. However, they can not directly be applied to our task without labelled data, as they are supervised. Our USGAN is able to learn from raw textual evidence and document-expert associations in an unsupervised way and thus does not require any manual relevance judgements between topics and experts.

Additionally, most unsupervised expert retrieval algorithms are based on language models [15]. However, these unsupervised language model-based algorithms need to construct a language model for every document in the collections, and thus lack efficient query capabilities for large document collections, as each query term needs to be exactly matched against every document. They also suffer from term mismatch problem, which occurs due to the inability of widely used maximum-likelihood functions in the algorithms

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313625

and the different representations of queries and expert documents to describe the same concepts [50]. To address the problems, we represent all the words and candidate experts as semantic vectors obtained from our unsupervised USGAN algorithm such that we can avoid constructing language models for all documents in the collections and the the semantic similarities between words and experts can be effectively measured.

The purposes of this work are not only for improving the performance of unsupervised expert retrieval algorithms, but also, more importantly, for providing a new insight into training GAN and its variants, i.e., via unsupervised learning for a number of information retrieval applications. Our contributions can be summarized as:

(1) We propose unsupervised generative adversarial network algorithm that is a minimax game consisting of a discriminative network and a generative network for expert retrieval task.

(2) Our USGAN is semantic and is able to semantically measure the similarities between the representations of words and experts.

(3) To our knowledge, we are the first attempt to apply GAN framework *in an unsupervised way* to tackle the retrieval problems in information retrieval.

(4) Through USGAN as an example, we would provide insight and inspiration on how to extend supervised GAN model and its variants in unsupervised ways to tackle other tasks where labelled data is limited or missing.

(5) We systematically analyze the proposed algorithms and find that we achieve better performance compared to the state-of-the-art unsupervised algorithms.

The remainder of the paper is organized as: Section 2 discusses related work; Section 3 details the research problem; Section 4 describes the proposed USGAN model; Section 5 describes our experimental setup; Section 6 is devoted to our experimental results and we conclude the paper in Section 7.

2 RELATED WORK

In this section, we only discuss the most related work: previous generative adversarial networks, expert retrieval algorithms, and semantic models.

2.1 Generative Adversarial Networks

Generative adversarial networks are a class of GAN-based methods for learning from labelled data based on minimax game theory [20] that consists of a generative network G and a discriminative network D. The goal of GAN-based methods is to train a generator network $G(\boldsymbol{z};\boldsymbol{\theta}^{(G)})$ that produces samples from the data distribution, $p_{\text{data}}(\boldsymbol{x})$, by transforming vectors of noise $\boldsymbol{z}$ as $\boldsymbol{x} = G(\boldsymbol{z};\boldsymbol{\theta}^{(G)})$. The training signal for G is provided by a discriminator network D that is trained to distinguish between samples from the generator distribution and those from ground truth (real) data. The generator network G in turn is then trained to fool the discriminator into accepting its outputs as being real. After the first GAN model is introduced, many of its recent variants have been proposed and successfully applied to many applications. E.g., Boundary equilibrium GAN is proposed in [11], where a novel equilibrium method is applied for balancing adversarial networks and auto-encoder is applied as the discriminator. Rather than proposing a new GAN-based algorithm, IRGAN [51] assumes that ground truth data is available for training and directly applies GAN to document retrieval task. Energy-based GAN [58] views the discriminator as an energy function that attributes low energies to the regions near the data manifold and higher energies to other regions. [47] introduced a specific conditioning of convolutional GAN on texture and shape elements for generating fashion design images. Some work aims at improving training techniques for GAN-based algorithms. E.g., Wasserstein GAN [2] tries to improve the stability of learning in GAN, gets rid of problems like model collapse, and provides learning curves for debugging and hyper-parameter searches. In [46], a variety of new architectural features and training procedures are proposed and applied into GAN framework. Previous work has shown that GAN-based algorithms can perform better than most traditional deep learning models such as LSTM (Long Short Term Memory networks) and CNN (Convolutional Neural Networks) [19].

Surprisingly, as far as we know little attention has paid attention to training GAN and its variants *without labelled data* for tasks in information retrieval, and the lack of training data in previous supervised GAN based algorithms would greatly hinder their success in the tasks. To the best of our knowledge, ours is the first attempt to train GAN in an unsupervised way for expert retrieval and the first attempt to apply GAN to applications in information retrieval via unsupervised ways.

2.2 Expert Retrieval

After the launch of expert retrieval task at TREC 2005 enterprise track [14], a number of supervised and unsupervised algorithms for the task have been proposed, which can be categorized into either supervised or unsupervised ones. Supervised algorithms include the relevance-based expert retrieval learning algorithm [18], learning to find experts [52]. Of special relevance to us are the unsupervised algorithms. The most well-known unsupervised ones are the profile-centric (Model 1) and document-centric (Model 2) expert finding models [4, 8], which focus on raw textual evidence without incorporating collection-specific information, e.g., query modeling, document importance or document structure. Both the Model 1 and Model 2 expert finding algorithms are language retrieval models, and they differ each other by the way of computing the probabilities of a candidate having expertise on a given topic. Other unsupervised ones include thread-based model [55] where expert finding is performed in the context of social media, language model for finding bloggers as experts [7] in the context of blog documents, and shallow log-linear model [50] that learns distributed word representations in an unsupervised way for the expert finding task, probabilistic expert finding models [17] where candidate generation framework and topic generation framework are studied for the task, and language models for finding groups of experts [32, 34] that aim at retrieving a ranked list of groups of experts given a topic. All these supervised and unsupervised methods are not based on GAN, and except the shallow log-linear one they are non-semantic.

To better provide insight in how our unsupervised semantic GAN-based algorithm improves performance of expert retrieval, we follow the setting applied in many unsupervised expert retrieval methods, i.e., avoid explicit feature engineering and the incorporation of external evidence in our USGAN expert retrieval algorithm.

To the best of our knowledge, we are the first attempt to apply GAN techniques to tackle the task of expert retrieval.

2.3 Semantic Models

The mismatches between queries and documents pose one of the most critical challenges in many information retrieval tasks [29, 30, 40, 41], and a number of topic models have been proposed to alleviate the mismatch challenge. Topic models provide a suite of semantic algorithms to discover hidden thematic structure in a collection of documents. A topic model takes a set of documents as input, and discovers a set of "latent topics"—recurring semantic themes that are discussed in the collection—and the degree to which each document exhibits those semantic topics [12]. Since the well-known topic models, PLSI (Probabilistic Latent Semantic Indexing) [22] and LDA (Latent Dirichlet Allocation) [12] were proposed, various topic models and their applications in information retrieval have been widely studied. The author topic model [45] has been proposed to uncover latent semantic topics of authors; each author is associated with a multinomial distribution over topics and each topic is associated with a multinomial distribution over words. In [53], LDA-based document models have been proposed to perform semantic matching of documents and queries. Topic models were proposed for clustering short texts [39] and users [40] in streams, streaming short text diversifications [31, 38], diversified data fusion of rank lists [31, 33, 35–37]. In [56], a comparative study of utilizing topic models for document retrieval was provided. Besides topic models, other researchers tried to alleviate the mismatches between queries and documents in other ways. For instance, in [23] semantic matching between queries and documents was performed by leveraging click-through data optimizing for web document ranking. Deep learning models, e.g., position-aware representations for relevance matching model [24], deep relevance matching model [21], shallow log-linear model [50] and multiple instance deep learning model [13], have been proposed to alleviate the mismatch problem between queries and documents.

Except the shallow log-linear model [50], all the existing semantic models are for mismatch problem between queries and documents only but not for queries and candidate experts. Unlike all of the previous models, we tackle the problem of mismatch problem among queries, documents and candidate experts via an unsupervised GAN-based semantic model. To the best of our knowledge, we are the first attempt to address the semantic mismatch problem via an unsupervised GAN method.

3 PROBLEM FORMULATION

We aim at addressing the expert retrieval task defined at the TREC 2005–2008 enterprise tracks [3, 9, 14, 48] in an unsupervised way: given a topic (a topic is also called a "query" containing a set of words in the tracks), a set of candidate experts and the heterogeneous enterprise document corpora, identify a rank list of experts who have expertise on the topic. The expert retrieval algorithm is essentially a function f that satisfies the following:

$$q, \mathcal{E}, \mathcal{D}, \xrightarrow{f} \mathcal{L},$$

where $q = \{v_1, \ldots, v_{|q|}\}$ is a query (also called topic at the tracks), $\mathcal{E}$ is a set of candidate experts, $\mathcal{D}$ is an organization's heterogeneous document corpora, and $\mathcal{L}$ is the final rank list of the experts in response to the query (experts with larger retrieval probabilities are ranked higher in the final rank list). Here v is a word from a vocabulary $\mathcal{V}$ and $|q|$ is the length of the query q.

4 METHOD

In this section, we detail our proposed **U**nsupervised **S**emantic **G**enerative **A**dversarial **N**etworks (USGAN) that aim at addressing the expert finding problem.

4.1 The Minimax Game

Our proposed USGAN for expert retrieval is an unsupervised, semantic minimax two-player game that consists of a discriminative model $D_{\boldsymbol{\phi}}$ with parameter $\boldsymbol{\phi}$ and a generative model $G_{\boldsymbol{\theta}}$ with parameter $\boldsymbol{\theta}$. The generative model would try to generate (in our task, select) relevant experts to the query that look like the ground-truth relevant experts and therefore could fool the discriminative model, whereas the discriminative model would try to draw a clear distinction between the ground-truth experts and the generated ones made by its opponent generative model. Competition in this game drives both models to improve their expert retrieval abilities until the discriminative model is indistinguishable from the experts selected from the generative model.

Without loss of generality, let's suppose we have N queries. Inspired by many GAN-based algorithms [20], we can simply define our $D_{\boldsymbol{\phi}}$ and $G_{\boldsymbol{\theta}}$ and let them play the following two-player minimax game:

$$\begin{aligned} J^{G_{\boldsymbol{\theta}^*}, D_{\boldsymbol{\phi}^*}} &= \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \sum_{n=1}^{N} \Big(\mathbb{E}_{e \sim p_{GT}(e|q_n)} [\log D_{\boldsymbol{\phi}}(e \mid q_n)] + \\ &\qquad \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}|q_n)} \Big[\log \Big(1 - D_{\boldsymbol{\phi}} \left(G_{\boldsymbol{\theta}}(\mathbf{z} \mid q_n) \mid q_n \right) \Big) \Big] \Big), \\ &= \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \sum_{n=1}^{N} \Big(\mathbb{E}_{e \sim p_{GT}(e|q_n)} [\log D_{\boldsymbol{\phi}}(e \mid q_n)] + \\ &\qquad \mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|q_n)} \Big[\log \Big(1 - D_{\boldsymbol{\phi}} \left(e \mid q_n \right) \Big) \Big] \Big), \end{aligned} \tag{1}$$

where e is a candidate expert, $p_{GT}(e \mid q_n)$ is the ground truth distribution of experts to the n-th query q_n, and $\mathbf{z}$ and $p_{\mathbf{z}}(\mathbf{z} \mid q_n)$ are a noise vector and its corresponding distribution, respectively. As shown in (1), we directly represent the generator $G_{\boldsymbol{\theta}}$ as $p_{\boldsymbol{\theta}}(e \mid q)$ (see the comparison between the expectation term in the second line of (1) and the expectation term in the last line of (1)), and according to many supervised discriminative expert retrieval algorithms [18], $D_{\boldsymbol{\phi}}(e_i \mid q_n)$ in (1) can be defined as:

$$D_{\boldsymbol{\phi}}(e_i \mid q_n) = p_{\boldsymbol{\phi}}(r = 1|e_i, q_n)^{r_{in}} p_{\boldsymbol{\phi}}(r = 0|e_i, q_n)^{1-r_{in}}, \tag{2}$$

where $p_{\boldsymbol{\phi}}(r = 1 \mid e_i, q_n)$ and $p_{\boldsymbol{\phi}}(r = 0 \mid e_i, q_n)$ are the probabilities of candidate e_i having and not having expertise on topic q_n, respectively, which can be directly obtained by the ground truth distribution $p_{GT}(e|q_n)$, and $r_{in} \in \{0, 1\}$ is the ground truth relevance score of candidate e_i having expertise on the topic q_n with $r_{in} = 1$ indicating that e_i is relevant to (has expertise on) the query q_n and $r_{in} = 0$ indicating the opposite.

Unfortunately, without ground truth distribution $p_{GT}(e \mid q_n)$ we can not play the minimax game defined in (1) to get the optimal discriminative model $D_{\boldsymbol{\phi}^*}$ and the optimal generative model $G_{\boldsymbol{\theta}^*}$.

Accordingly, we transfer (1) to the following unsupervised two-player minimax game:

$$J^{G_{\theta^*}, D_{\phi^*}} = \min_{\theta} \max_{\phi} \sum_{n=1}^{|\mathcal{S}|} \Big(\mathbb{E}_{e \sim p_{\phi}(e|s_n)} [\log D_{\phi}(e \mid s_n)] + \mathbb{E}_{e \sim p_{\theta}(e|s_n)} \left[\log \left(1 - D_{\phi} \left(e \mid s_n \right) \right) \right] \Big), \quad (3)$$

where $\mathcal{S} = \{s_n\}_{n=1}^{|\mathcal{S}|}$ is a set of word sequences (query can be regarded as a word sequence as well) extracted from all documents in $\mathcal{D}$ with $s_n = \{v_1, \ldots, v_{|s_n|}\}$ being the n-th word sequence, and the size of the sequence set is $|\mathcal{S}|$, i.e., the total number of word sequences in the set $\mathcal{S}$. Here $|s_n|$ is the size of the n-th sequence, i.e., the total number of words in the sequence s_n. Obviously, unlike queries that are manually defined in the ground truth, our set of word sequences $\mathcal{S}$ can be obtained by an unsupervised way. All the words in the sequences are supposed from a vocabulary $\mathcal{V}$. Note that both generative and discriminative networks with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively, are still modeled in (3), with the generative networks directly generate/select an expert from the distribution $p_{\theta}(e \mid s)$. Later in this section, we will model each sequence as m-grams extracted from documents.

As we do not have ground truth during unsupervised learning, without loss of generality, as can be seen in (3) we replace N in (1), the number of ground truth queries, to be $|\mathcal{S}|$, and $p_{GT}(e \mid q_n)$ in (1) to be $p_{\phi}(e \mid s_n)$, i.e., the probability of a candidate expert e having expertise on the topic underlying the sequence s_n, respectively. In what follows, we present how we define $p_{\phi}(e_i \mid s_n)$, $p_{\theta}(e_i \mid s_n)$ and D_{ϕ} in (3) via an unsupervised way, where we do not require topic-candidate ground truth assessments and manually defined queries for training. Recall that in (1) we directly represent G_{θ} as $p_{\theta}(e \mid q)$, and thus we do not need to redefine G_{θ} at all.

Definition of $p_{\phi}(e_i \mid s_n)$. According to unsupervised language model, we closely follow the work in [50] and rank all the candidate experts in $\mathcal{E}$ by the conditional probability of a candidate expert e_i given a topic/sequence $s_n = \{v_1, \ldots, v_{|s_n|}\}$:

$$p_{\phi}(e_i \mid s_n) = \frac{1}{Z_1} \widetilde{p}_{\phi}(e_i \mid v_1, \ldots, v_{|s_n|}) = \frac{1}{Z_1} \prod_{j=1}^{|s_n|} p_{\phi}(e_i \mid v_j)$$

$$\overset{\text{rank}}{=} \frac{1}{Z_1} \exp\left(\sum_{j=1}^{|s_n|} \log \left(p_{\phi}(e_i \mid v_j) \right) \right), \quad (4)$$

where $\widetilde{p}_{\phi}(e_i | v_1, \ldots, v_{|s_n|})$ is the unnormalized score, $Z_1 = \sum_{i=1}^{|\mathcal{E}|} \exp(\sum_{j=1}^{|s_n|} \log(p_{\phi}(e_i \mid v_j)))$ is the a normalization term (The transformation to log-space in (4) is a well-known trick to prevent floating point underflow [43]), $\overset{\text{rank}}{=}$ denotes that ranking according to the former equation is equivalent to ranking according to the latter one, and $p_{\phi}(e_i \mid v_j)$ is the probability of a candidate having expertise on the topic specific to a word $v_j \in \mathcal{V}$, which can be defined as a log-linear learning model:

$$p_{\phi}(e_i | v_j) = \frac{1}{Z_2} \exp(\mathbf{e}_i^{\phi} \cdot \mathbf{v}_j^{\phi \intercal} + b_i^{\phi}), \quad (5)$$

where $\mathbf{e}_i^{\phi}$ and $\mathbf{v}_j^{\phi}$ are the expert e_i's and the word v_j's $1 \times c$ semantic representation vectors with c being the size of the vectors, respectively, b_i^{ϕ} is a bias for the candidate expert e_i, $\mathbf{v}^{\phi \intercal}$ is the transpose of $\mathbf{v}^{\phi}$, and $Z_2 = \sum_{i=1}^{|\mathcal{E}|} \exp(\mathbf{e}_i \cdot \mathbf{v}_j^{\top} + b_i)$ is the normalization term. Obviously, the semantic representations of all the candidate experts, $\{\mathbf{e}_i^{\phi}\}_{i=1}^{|\mathcal{E}|}$, the semantic representations of all the words, $\{\mathbf{v}_j^{\phi}\}_{j=1}^{|\mathcal{V}|}$, and the corresponding biases $\{b_i^{\phi}\}_{i=1}^{|\mathcal{E}|}$ constitute the parameter $\boldsymbol{\phi}$ of our discriminative model D_{ϕ}, i.e., $\boldsymbol{\phi} = \left\{ \{\mathbf{e}_i^{\phi}\}_{i=1}^{|\mathcal{E}|}, \{\mathbf{v}_j^{\phi}\}_{j=1}^{|\mathcal{V}|}, \{b_i^{\phi}\}_{i=1}^{|\mathcal{E}|} \right\}$. Note that we define a neural network in (5) with a log-linear model only. Adding more layers into the neural network defined in (5) is possible. But our experiments found that the shallow log-linear model defined in (4) performs well-enough in most cases and adding more layers will require more training times and lose the transparency of the model. Other neural networks such as CNN for defining $p_{\phi}(e_i \mid s_n)$ are possible, but we want to be focused and leave these as future work.

Definition of $p_{\theta}(e_i \mid s_n)$. To focus more on our USGAN, we define $p_{\theta}(e_i \mid s_n)$ for our generative model in such a way that is the same as that in our discriminative model, i.e., via (4) and (5), but with different semantic parameters $\{\mathbf{e}_i^{\theta}\}_{i=1}^{|\mathcal{E}|}$, $\{\mathbf{v}_j^{\theta}\}_{j=1}^{|\mathcal{V}|}$ and $\{b_i^{\theta}\}_{i=1}^{|\mathcal{E}|}$. Then, the parameter $\boldsymbol{\theta}$ in our generative model is set to be $\boldsymbol{\theta} = \left\{ \{\mathbf{e}_i^{\theta}\}_{i=1}^{|\mathcal{E}|}, \{\mathbf{v}_j^{\theta}\}_{j=1}^{|\mathcal{V}|}, \{b_i^{\theta}\}_{i=1}^{|\mathcal{E}|} \right\}$. Again, to keep focused, we leave using alternative neural networks for generative model in our USGAN as future work.

Definition of D_{ϕ}. Before we follow that in [50] and define our final D_{ϕ}, we construct a pseudo ground truth probability of candidate e_i having expertise on a document $d_k \in \mathcal{D}$, i.e., $p_{\widetilde{GT}}(e_i \mid d_k)$, based on document-candidate associations by an unsupervised way:

$$p_{\widetilde{GT}}(e_i \mid d_k) = \begin{cases} 0, & e_i \notin \mathcal{E}_{d_k}, \\ \frac{1}{|\mathcal{E}_{d_k}|}, & e_i \in \mathcal{E}_{d_k}, \end{cases} \quad (6)$$

where $\mathcal{E}_{d_k}$ and $|\mathcal{E}_{d_k}|$ are a set of candidates whose names or email addresses are observed in the document d_k and the total number of the observed candidates in the document d_k (i.e., the size of the set), respectively. For each document $d \in \mathcal{D}$, we extract a set of m-grams where m remains fixed during our unsupervised learning. Let $\mathcal{M}_{d_k}$ denote the set of the m-grams extracted from d_k with $d_k^{(l)} \in \mathcal{M}_{d_k}$ being the l-th m-grams. Let $\mathbf{p}_{\widetilde{GT}}(e_i \mid d_k) = \{p_{\widetilde{GT}}(e_i \mid d_k^{(l)})\}_{l=1}^{|\mathcal{M}_{d_k}|}$, where $|\mathcal{M}_{d_k}|$ is the size of the m-grams (i.e., word sequences) set extracted from d_k and let $p_{\widetilde{GT}}(e_i \mid d_k^{(l)}) = p_{\widetilde{GT}}(e_i \mid d_k)$ which can be obtained by (6). We also denote $\mathbf{p}_{\phi}(e_i \mid d_k)$ to be $\mathbf{p}_{\phi}(e_i \mid d_k) = \{p_{\phi}(e_i \mid d_k^{(l)})\}_{l=1}^{|\mathcal{M}_{d_k}|}$. Here $p_{\phi}(e_i \mid d_k^{(l)}) = p_{\phi}(e_i \mid v_1^{d_k^{(l)}}, \ldots, v_m^{d_k^{(l)}})$ can be obtained by (4) with $v_x^{d_k^{(l)}}$ being the x-th word in the m-gram $d_k^{(l)}$. Accordingly, we define our discriminative model $D_{\phi}(e_i \mid d_k)$ based on the cross-entropy $C\left(\mathbf{p}_{\widetilde{GT}}(e_i \mid d_k), \mathbf{p}_{\phi}(e_i \mid d_k)\right)$ in an unsupervised way as follows:

$$
\begin{aligned}
D_{\boldsymbol{\phi}}(e_i \mid d_k) =& \frac{1}{Z_3} \frac{|d_{\max}|}{|d_k|} C\left(\mathbf{p}_{\widetilde{GT}}(e_i \mid d_k), \mathbf{p}_{\boldsymbol{\phi}}(e_i \mid d_k)\right) \\
&+ \epsilon \left(\|\mathbf{e}_i^{\boldsymbol{\phi}}\|_2^2 + \frac{1}{|\mathcal{M}_{d_k}|} \sum_{j=1}^{|\mathcal{M}_{d_k}|} \|\mathbf{v}_j^{\boldsymbol{\phi}}\|_2^2 \right) \\
= -& \frac{|d_{\max}|}{Z_3 \cdot |d_k| \cdot |\mathcal{M}_{d_k}|} \sum_{l=1}^{|\mathcal{M}_{d_k}|} \left(p_{\widetilde{GT}}(e_i \mid d_k^{(l)}) \cdot \log p_{\boldsymbol{\phi}}(e_i \mid d_k^{(l)}) \right) \\
&+ \epsilon \left(\|\mathbf{e}_i^{\boldsymbol{\phi}}\|_2^2 + \frac{1}{|\mathcal{M}_{d_k}|} \sum_{j=1}^{|\mathcal{M}_{d_k}|} \|\mathbf{v}_j^{\boldsymbol{\phi}}\|_2^2 \right), \qquad (7)
\end{aligned}
$$

where $\|\mathbf{x}\|_2$ is the 2-norm of a vector $\mathbf{x}$, ϵ is a free regularization parameter, $d_{\max} = \arg\max_{d \in \mathcal{D}} |d|$ is the longest document in the collection, $|d_{\max}|$ and $|d_k|$ are the length of the corresponding documents, respectively, and Z_3 is a normalization constant that forces $D_{\boldsymbol{\phi}}(e \mid d)$ to be within the range $(0, 1]$, which can be defined to be the maximal score of $D_{\boldsymbol{\phi}}(e \mid d)$ before the normalization.

4.2 Optimizing the Discriminative Model

The objective of the discriminative model is to maximize the log-likelihood of correctly distinguishing the candidate experts from $p_{\boldsymbol{\phi}}(e \mid s)$ and those generated from $p_{\boldsymbol{\theta}}(e \mid s)$. As (7) is built on (4) and thus maximizing (7) for all candidates and documents means to maximizing (4) for all the candidates and documents in collection, and with the candidate experts sampled from the current optimal generative model $p_{\boldsymbol{\theta}^*}(e \mid s)$, we can obtain the optimal parameters for the discriminative model defined in (3) as:

$$
\begin{aligned}
\boldsymbol{\phi}^* =& \arg\max_{\boldsymbol{\phi}} \sum_{k=1}^{|\mathcal{D}|} \Bigg(\mathbb{E}_{e \sim p_{\boldsymbol{\phi}}(e|d_k)}[\log D_{\boldsymbol{\phi}}(e \mid d_k)] + \\
& \mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k)) \right] \Bigg) \\
=& \arg\max_{\boldsymbol{\phi}} \sum_{k=1}^{|\mathcal{D}|} \Bigg(\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \log D_{\boldsymbol{\phi}}(e_i \mid d_k) + \\
& \mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k)) \right] \Bigg). \qquad (8)
\end{aligned}
$$

As can be seen in (7), $D_{\boldsymbol{\phi}}(e \mid d)$ is differentiable with respect to the parameter $\boldsymbol{\phi}$, and thus we can obtain the optimal parameter $\boldsymbol{\phi}^*$ in (8) by stochastic gradient descent. For convenient discussion in later part of the paper, we denote $\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \log D_{\boldsymbol{\phi}}(e_i \mid d_k) + \mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k) \right]$ in (8) as $J^{D_{\boldsymbol{\phi}}(d_k)}$ and thus (8) can be simplified as $\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \sum_{k=1}^{|\mathcal{D}|} J^{D_{\boldsymbol{\phi}}(d_k)}$.

4.3 Optimizing the Generative Model

In contrast, the objective of the generative model $G_{\boldsymbol{\theta}}$ with the form of $p_{\boldsymbol{\theta}}(e \mid d)$ is to minimize the log-likelihood of $1 - D_{\boldsymbol{\phi}}$ such that the model can fit the underlying relevance distribution over candidate experts, $p_{\widetilde{GT}}(e \mid d)$. The generative model samples candidates from the whole candidate expert set $\mathcal{E}$ based on the distribution $p_{\boldsymbol{\theta}}(e \mid d)$ and tries to fool the discriminative model $D_{\boldsymbol{\phi}}$.

It is worth mentioning that unlike most GAN-based models [1, 58], we let our generative model $G_{\boldsymbol{\theta}}$ directly generate known candidate experts but not their features, because our work here intends to directly select relevant experts from a given candidate set $\mathcal{E}$. Note that as presented in (1), it is feasible to first generate features as denoted by $\mathbf{z}$ and then generate candidate experts from $G_{\boldsymbol{\theta}}(\mathbf{z} \mid s)$ by USGAN. However, to be focused, we leave this as future work.

After we obtain the optimal parameter $\boldsymbol{\phi}^*$ from (8), we then optimize the generative model $G_{\boldsymbol{\theta}}$ via performing the following minimization:

$$
\begin{aligned}
\boldsymbol{\theta}^* =& \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{|\mathcal{D}|} \Bigg(\underbrace{\mathbb{E}_{e \sim p_{\boldsymbol{\phi}}(e|d_k)}[\log D_{\boldsymbol{\phi}}(e \mid d_k)]}_{\text{Partial derivative of this term w.r.t } \boldsymbol{\theta} \text{ is } 0} + \\
& \mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k) \right] \Bigg) \\
=& \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{|\mathcal{D}|} \underbrace{\mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k) \right]}_{\text{Denoted this term as } J^{G_{\theta}(d_k)}}, \qquad (9)
\end{aligned}
$$

where we denote the term $\mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k) \right]$ as $J^{G_{\boldsymbol{\theta}}(d_k)}$ and thus it can be simplified to be $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{|\mathcal{D}|} J^{G_{\boldsymbol{\theta}}(d_k)}$. As the sampling $e \sim p_{\boldsymbol{\theta}}(e \mid d_k)$ is discrete in the objective function (9) and unlike many GAN-based algorithms [20], we can not simply optimize our $\boldsymbol{\theta}$ by gradient descent. We utilize policy gradient based reinforcement learning (REINFORCE) [49, 54, 57] for obtaining the optimal $\boldsymbol{\theta}^*$ in (9), and the gradient is derived as:

$$
\begin{aligned}
& \nabla_{\boldsymbol{\theta}} J^{G_{\boldsymbol{\theta}}(d_k)} \qquad (10) \\
&= \nabla_{\boldsymbol{\theta}} \left(\mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k) \right] \right) \\
&= \sum_{i=1}^{|\mathcal{E}|} \nabla_{\boldsymbol{\theta}} \left(p_{\boldsymbol{\theta}}(e_i \mid d_k) \log(1 - D_{\boldsymbol{\phi}}(e_i \mid d_k)) \right) \\
&= \sum_{i=1}^{|\mathcal{E}|} p_{\boldsymbol{\theta}}(e_i \mid d_k) \nabla_{\boldsymbol{\theta}} \left(\log p_{\boldsymbol{\theta}}(e_i \mid d_k) \log(1 - D_{\boldsymbol{\phi}}(e_i \mid d_k)) \right) \\
&= \mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\nabla_{\boldsymbol{\theta}} \left(\log p_{\boldsymbol{\theta}}(e_i \mid d_k) \log(1 - D_{\boldsymbol{\phi}}(e_i \mid d_k)) \right) \right] \\
&\simeq \frac{1}{H} \sum_{i=1}^{H} \nabla_{\boldsymbol{\theta}} \left(\log p_{\boldsymbol{\theta}}(e_i \mid d_k) \log(1 - D_{\boldsymbol{\phi}}(e_i \mid d_k)) \right),
\end{aligned}
$$

where in the last step we sample H candidates from the current version of generative model $p_{\boldsymbol{\theta}}(e \mid d_k)$ to approximate the partial derivative of $J^{G_{\boldsymbol{\theta}}(d_k)}$ with regards to the parameter $\boldsymbol{\theta}$. According to reinforcement learning [49, 54], the term $\log(1 - D_{\boldsymbol{\phi}}(e_i \mid d_k))$ in the last step of (10) acts as the reward for the policy $p_{\boldsymbol{\theta}}(e \mid d_k)$ taking an action e_i in the environment d_k. Other ways, e.g., categorical reparameterization with Gumbel-Softmax [25], to derive the gradient $\nabla_{\boldsymbol{\theta}} J^{G_{\boldsymbol{\theta}}(d_k)}$ are possible, but we leave this for future investigation.

To reduce variance during the REINFORCE learning, we replace the reward term $\log(1 - D_{\boldsymbol{\phi}}(e_i \mid d_k))$ by its advantage function as:

$$
\log(1 - D_{\boldsymbol{\phi}}(e_i \mid d_k)) - \mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}(e \mid d_k)) \right],
$$

where the term $\mathbb{E}_{e \sim p_{\boldsymbol{\theta}}(e|d_k)} \left[\log(1 - D_{\boldsymbol{\phi}}\,(e \mid d_k)) \right]$ acts as the baseline function in policy gradient based reinforcement learning [49].

4.4 Overview, Example and Discussion

Overview of USGAN. The overview of our proposed unsupervised semantic USGAN expert retrieval model is shown in Algorithm 1. We initialize the semantic representations of all words, $\{\mathbf{v}_j^{\phi}\}_{j=1}^{|\mathcal{V}|}$ and $\{\mathbf{v}_j^{\theta}\}_{j=1}^{|\mathcal{V}|}$, in both the discriminative and generative models, with pre-trained word representations trained on Wikipedia 2014 dataset [1] [44], respectively. The dimension of both words' and candidates' semantic representations is set to be $c = 300$. We sample initial vectors uniformly in the range $[-\frac{1}{\sqrt{300}}, \frac{1}{\sqrt{300}}]$ for all the candidates as their initial semantic representations, and let all the initial biases be 0. (step 1 in Algorithm 1). We pre-train our $\boldsymbol{\phi}$ by maximizing (7) using batched gradient descent. After the pre-training of $\boldsymbol{\phi}$, we let $\boldsymbol{\theta} = \boldsymbol{\phi}$. (step 2 in Algorithm 1). We iteratively perform the minimax game where we update the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ at each iteration and terminate the game once it converges. Obviously, we can perform the unsupervised training for our USGAN offline, whereas the USGAN with optimal parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ after training can be directly applied online for expert retrieval.

After we obtain the optimal parameters $\boldsymbol{\phi}^*$ of the discriminative neural-network and $\boldsymbol{\theta}^*$ of the generative neural-network in USGAN, given a query q, we produce a rank list of the candidate experts $\mathcal{L}$ by their relevance probabilities to the query, i.e., either by $p_{\boldsymbol{\phi}^*}(e \mid q)$ with parameter $\boldsymbol{\phi}^*$ of the discriminative model or by $p_{\boldsymbol{\theta}^*}(e \mid q)$ with parameter $\boldsymbol{\theta}^*$ of the generative model. See (4) for computing $p_{\boldsymbol{\phi}^*}(e \mid q)$ and $p_{\boldsymbol{\theta}^*}(e \mid q)$. Obviously, the unsupervised training for USGAN can be performed offline, whereas the USGAN with optimal parameters $\boldsymbol{\phi}^*$ and $\boldsymbol{\theta}^*$ after the training can be directly applied online for the expert retrieval task.

An Example. Here we provide an example to further illustrate the underlying idea of our proposed USGAN model. We are given a corpora of documents (any formats, e.g., Code docs, Emails, web pages, Wiki web pages etc.), a set of test queries, a set of candidate experts, and the ground truth. Suppose the corpora only contains two Wiki documents: d_1 and d_2, the content of which are: d_1={"*Turing was influential in development of theoretical computer science, providing a formalization of the concepts of algorithm and computation.*"}, d_2={"*Washington was an American statesman who served as the first president of the United States.*"}, respectively, and no other information of the documents is available. We further suppose there is only one test query, q_1={"algorithm"}, and the ground truth for evaluation is: {q_1–Turing–1; q_1–Washington–0} where the "1" indicates that the expert is relevant to the query; whereas "0" indicates non-relevant. Thus, the ground truth denotes that Turing is an expert on topic q_1, whereas Washington is not an expert on topic q_1. The goal of our USGAN is to obtain embeddings (semantic vectors) of both the candidate experts and the words in the corpora, where we assume that the ground truth is not available during training (the ground truth can only be used for evaluation). According to (6), we have $p_{\widetilde{GT}}(\text{Turing} \mid d_1) = 1$, $p_{\widetilde{GT}}(\text{Washington} \mid d_1) = 0$, $p_{\widetilde{GT}}(\text{Turing}|d_2) = 0$, $p_{\widetilde{GT}}(\text{Washington}|d_2) = 1$. All of these probabilities are obtained in an unsupervised way (just identify if the names can be observed in the documents) and are used as the pseudo relevance signals during the unsupervised training of our USGAN. We use m-grams during the training. Suppose $m = 1$ and thus we have probabilities (signals) of each word in the document for the unsupervised training, e.g., $p_{\widetilde{GT}}(\text{Turing} \mid \text{"computer"}) = 1$, (as $p_{\widetilde{GT}}(\text{Turing} \mid d_1) = 1$ and the word "computer" is within d_1), $p_{\widetilde{GT}}(\text{Turing}|\text{"algorithm"}) = 1$, etc. These signals will be used during the unsupervised training. Once the training is done, we will obtain the embeddings. Given a test query={"algorithm"}, we rank the experts by $p_{\boldsymbol{\phi}^*}(e \mid q)$ or $p_{\boldsymbol{\theta}^*}(e \mid q)$ computed by (4), which will result in a ranking, and will be evaluated by the ground truth.

Algorithm 1: Overview of USGAN for expert retrieval.

```
   Input  : $\mathcal{D}$, $\mathcal{E}$, $\epsilon$
   Output : Optimal $\boldsymbol{\phi}^*$ of $D_{\boldsymbol{\phi}^*}$, and Optimal $\boldsymbol{\theta}^*$ of $G_{\boldsymbol{\theta}^*}$
1  Initialize $\boldsymbol{\phi}$ of $D_{\boldsymbol{\phi}}$ and $\boldsymbol{\theta}$ of $G_{\boldsymbol{\theta}}$
2  Pre-train $p_{\boldsymbol{\phi}}(e \mid d)$ and $p_{\boldsymbol{\theta}}(e \mid d)$ with all documents in $\mathcal{D}$
     by (7)
3  repeat
4      for g-steps do
5          for $k = 1, \ldots, |\mathcal{D}|$ do
6              Sample top-$H$ candidate experts from the current
                 version of generative model $p_{\boldsymbol{\theta}}(e \mid d_k)$ for
                 document $d_k$
7          Update $\boldsymbol{\theta}$ in the generator $G_{\boldsymbol{\theta}}$, i.e., (9), by descending
             its stochastic gradient with REINFORCE learning:
             $\nabla_{\boldsymbol{\theta}} \sum_{k=1}^{|\mathcal{D}|} J^{G_{\boldsymbol{\theta}}(d_k)} = \sum_{k=1}^{|\mathcal{D}|} \nabla_{\boldsymbol{\theta}} J^{G_{\boldsymbol{\theta}}(d_k)}$
8      for d-steps do
9          for $k = 1, \ldots, |\mathcal{D}|$ do
10             Sample top-$H$ candidate experts from the current
                 version of generative model $p_{\boldsymbol{\theta}}(e \mid d_k)$ for
                 document $d_k$
11         Update $\boldsymbol{\phi}$ in the discriminative $D_{\boldsymbol{\phi}}$, i.e., (8), by
             ascending its stochastic gradient:
             $\nabla_{\boldsymbol{\phi}} \sum_{k=1}^{|\mathcal{D}|} J^{D_{\boldsymbol{\phi}}(d_k)} = \sum_{k=1}^{|\mathcal{D}|} \nabla_{\boldsymbol{\phi}} J^{D_{\boldsymbol{\phi}}(d_k)}$
12 until USGAN converges
```

Discussions. To our knowledge, there is only one GAN-based algorithm for information retrieval – IRGAN [51]. Our USGAN differs from IRGAN in at least four aspects: (1) The main difference between IRGAN and our USGAN lies in the fact that IRGAN requires ground truth data to train the model, which will hinder its applications in the scenarios that no labelled data is available, whereas our USGAN is an unsupervised GAN-based algorithm and can work without labelled data. (2) IRGAN directly defines its discriminative networks using ground truth data – the samples from the generative networks can be directly distinguished with the labelled data in the ground truth, whereas we define an unique discriminative function for the expert retrieval task in an unsupervised way (see (7)). (3) Our USGAN is semantic retrieval networks and in which we avoid feature extractions, whereas IRGAN requires to extract features for query-entity pairs for training the networks. (4) It is impossible to directly apply IRGAN to the task of expert retrieval even if we assume that the labelled data is available, as there is no direct connection between a relevant expert and a query. In addition, a state-of-the-art UESM (unsupervised, efficient and semantic expertise model) model [50] is just proposed, and has

[1] Publicly available from https://nlp.stanford.edu/projects/glove/.

shown its good performance on the expert retrieval task. Our USGAN differs from UESM in at least three aspects: (1) The strategy to obtain optimal parameters in USGAN is different compared to that in UESM. USGAN utilizes reinforcement learning to update the parameters; whereas UESM directly apply stochastic gradient descent to obtain them. (2) The frameworks are absolutely different. USGAN is a GAN-based algorithm, whereas UESM is a common neural network-based language model. (3) USGAN captures two parameters: $\boldsymbol{\phi}$ from its discriminative network and $\boldsymbol{\theta}$ from its generative network. The competitions between the two networks drive both networks to improve their semantic representations; whereas UESM is built on one network only and no competitions in it.

5 EXPERIMENTAL SETUP

In this section, we describe our experimental setup.

5.1 Research Questions

The research questions guiding the remainder of the paper are: **(RQ1)** How is the expert retrieval performance of USGAN compared to other state-of-the-art methods? **(RQ2)** What is the effect of the window size (m-grams) on the performance of our USGAN expert retrieval method? **(RQ3)** What is the effect of the sampling size (parameter H in (10)) on the performance of our USGAN? **(RQ4)** What is impact of number of iterations on the retrieval performance of our generative model $G_{\boldsymbol{\theta}}$ and discriminative model $D_{\boldsymbol{\phi}}$? **(RQ5)** Does our USGAN outperform the best unsupervised baseline method on each query?

5.2 Dataset

For evaluation purposes we use datasets made available at the 2005 and 2006 editions of the TREC enterprise tracks [14, 48]. Note that for both of these two years' tracks, the document collections [2] are the same but with different testing queries. The document collection contains a crawl of the World Wide Web Consortium (W3C) – a heterogenous document repository containing a mixture of document types, which include lists (from email forum), dev, www, esw, other, and people (personal homepages). Statistical information of the dataset is as follows: total number of documents: 331,037, average document length: 1,237.23, total number of candidates: 1092, total number of document-candidate associations: 200,939 (an association indicates that name or email address of a candidate expert appears in a document), average number of associations per document (Only documents with at least one association are considered): 2.14, average number of associations per candidate: 281.03, 50 test queries were created for the 2005 track with all of their ground truths provided, and 55 test queries were created for the 2006 track with only 50 of their ground truths provided. Other datasets, such as the CSIRO Enterprise Research Collection (CERC) dataset that was a dump of the intranet of Australia's national science agency and was built in the TREC 2007 and 2008 enterprise tracks [9] for testing and the employee dataset of the Tilburg University that consists of bi-lingual, heterogeneous documents [10], are impossible to be used for evaluating expert retrieval algorithms, because they are no longer publicly distributed. [3]

5.3 Baselines

As our goal is to improve performance of the expert retrieval task in an unsupervised way with no labelled data, we only consider unsupervised baselines for comparisons. As IRGAN is a supervised method, it is inappropriate to be taken as baseline. We make comparisons among our USGAN and the following unsupervised non-neural networks-based and neural networks-based algorithms:

Term Frequency and Inverse Document Frequency (TF-IDF). It weights a term by considering both its frequency and inverse document frequency in a given document [15].

Latent Dirichlet Allocation (LDA). This model [12] infers semantic topic distributions specific to each document and each word via the LDA model.

Author Topic Model (AuthorT). This model [45] infers semantic topic distributions specific to each candidate, each document and each word.

Profile-Centric Language Model (Model 1). It is an unsupervised, non-semantic language model for expert finding. It amasses all the term information from all the documents associated with the candidate for a given query and uses this to represent that candidate [4, 8].

Document-Centric Language Model (Model 2). It is an unsupervised, non-semantic language model for expert finding. It first selects documents associated with the candidate, using either the so-called document-centric or the candidate-centric language model. It then considers whether the query can be generated from those documents [4, 8].

Unsupervised, Efficient and Semantic Model (UESM). It is a state-of-the-art unsupervised, semantic, neural expert finding model based on neural networks, exclusively employs textual evidence from document-expert associations for learning representations of terms and experts in an unsupervised way [50].

In terms of the baselines, TF-IDF, LDA and AuthorT, we represent experts, documents and queries by their semantic vector spaces inferred from the corresponding models, and then adapt a generic framework proposed in [16] to rank experts in response to a query, respectively. For Models 1 & 2, we apply Dirichlet smoothing method with parameter β equal to the average document length [5]. We only include the state-of-the-art deep learning model, UESM, for comparisons. Traditional deep learning models, e.g., CNN, Recurrent Neural Network (RNN), LSTM and their state-of-the-art variants [19], are supervised learning algorithms, and thus they are inappropriate to be included as our baselines. There are some unsupervised deep learning models, but they mainly aim at encoding and decoding for generating semantic features in unsupervised ways and serve for different purposes [19]. Thus these unsupervised deep learning models can not be directly applied to expert finding task and used as our baselines.

Later for convenient discussion, we denote our methods that produce the retrieval results by the discriminative model $p_{\boldsymbol{\phi}^*}(e \mid q)$ and the generative model $p_{\boldsymbol{\theta}^*}(e \mid q)$ as USGAN$_D$ and USGAN$_G$, respectively.

5.4 Evaluation Metrics and Settings

The evaluation metrics used to assess the performance of the expert retrieval algorithms are the ones widely used in TREC enterprise

[2] Publicly available from http://trec.nist.gov/data/enterprise.html.

[3] Details can be found at http://trec.nist.gov/data/enterprise.html.

tracks [14, 48]: MAP (Mean Average Precision), R-Prec, MRR (Mean Reciprocal Rank) and P@k (Precision at k) [15]. R-Prec is the precision after R documents have been retrieved, where R is the total number of relevant documents for the query. We also use NDCG@k (Normalized Discounted Cumulative Gain at k [26]). For P@k and NDCG@k, we set k to 5 and 10, respectively, to align with the cut-off used in the TREC enterprise tracks.

For our USGAN (both USGAN$_D$ and USGAN$_G$) and the baseline UESM, we use a 70%/20%/10% split of the queries for unsupervised training, validation and test sets, respectively. We optimize the objectives during the unsupervised training using different values of the parameters, e.g., the size of m-grams in the methods, and the sampling size in USGAN. The best values are then chosen on the validation set, and evaluated on the test queries. The *g-steps* and *d-steps* epochs in USGAN (see Algorithm 1) keep running until the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ updated by the corresponding stochastic gradients converge, respectively. Other settings, e.g., the initialization for $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ etc., can be found in subsection 4.4. The train/validation/test splits are permuted until all the queries were chosen once for the test set. We repeat the experiments 10 times and report the average results. The statistical significance of the observed differences between the performance of two ranking algorithms across the queries is tested using a two-tailed paired t-test and is denoted using $^{\blacktriangle}$ (or $^{\blacktriangledown}$) for $\alpha = .01$, and $^{\vartriangle}$ (and $^{\triangledown}$) for $\alpha = .05$.

6 RESULTS AND ANALYSIS

In what follows, we discuss and analyze our experimental results and answer the research questions. Subsection 6.1 report the overall performance of USGAN and the baseline methods (**RQ1**); subsection 6.2 investigates how the size of the window (m-grams) impacts the performance of USGAN (**RQ2**); subsection 6.3 investigates the effect of the sampling size, i.e., the value of the parameter H in (10) on the performance of USGAN (**RQ3**); subsection 6.4 illustrates the impact of iterations on the retrieval performance of our generative and discriminative models (**RQ4**); finally, subsection 6.5 shows the performance of USGAN on different test queries (**RQ5**).

6.1 Overall of Experimental Results

RQ1: We compare the retrieval performance of our USGAN with the baseline methods on the TREC enterprise 2005 and 2006 tracks.

Table 1 reports the performance on all the evaluation metrics. For all these two years' tracks, we have the following findings from Table 1: (i) Both our discriminative model, USGAN$_D$, and our generative model, USGAN$_G$, are able to statistically significantly outperform all the baselines on all the metrics and in both of the two tracks, which confirms the effectiveness of our unsupervised generative adversarial networks for the expert retrieval task and the fact that utilizing the REINFORCE policy gradient can help to obtain optimal parameters during our unsupervised training in Algorithm 1. (ii) All the semantic methods, i.e., USGAN$_D$, USGAN$_G$, UESM, AuthorT, and LDA, outperform non-semantic baseline TF-IDF, which demonstrates that representing experts and words in semantic ways can help to improve the retrieval performance, compared to those directly representing them by words. (iii) Both USGAN$_D$ and USGAN$_G$ outperform the state-of-the-art semantic UESM model, which demonstrates the fact that semantic representations of experts and words obtained by GAN-based algorithms are better than those obtained by traditional deep learning models for the task of expert retrieval. (iv) USGAN$_D$ works somewhat better than USGAN$_G$, which demonstrates that our discriminative model outperforms our generative model. The main reason is because we can directly update the parameter $\boldsymbol{\phi}$ in USGAN$_D$ by stochastic gradient descent, whereas we can only approximately update the parameter $\boldsymbol{\theta}$ in USGAN$_G$ by REINFORCE learning where we need to select top-H discrete candidates.

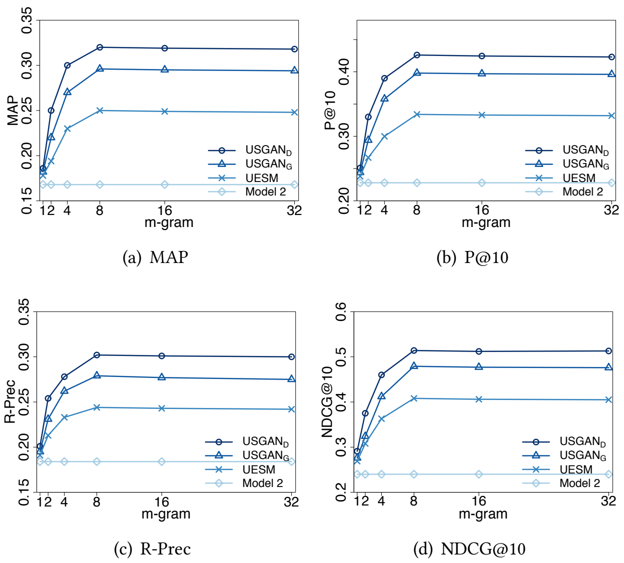
Table 1: Retrieval performance of USGAN and the baselines on MAP, R-Prec, MRR, P@5, 10, NDCG@5, 10, respectively. Statistically significant differences between USGAN$_G$ and best baseline model, UESM, and between USGAN$_D$ and USGAN$_G$ are marked in the upper right hand corner of USGAN$_G$'s and USGAN$_D$'s scores, respectively.

					P@		NDCG@	
		MAP	R-Prec	MRR	5	10	5	10
2005	TF-IDF	.147	.165	.520	.188	.196	.199	.202
	LDA	.152	.170	.500	.200	.204	.207	.209
	AuthorT	.158	.171	.480	.216	.210	.227	.220
	Model 1	.165	.192	.560	.228	.218	.231	.226
	Model 2	.168	.184	.600	.236	.228	.247	.240
	UESM	.250	.244	.760	.420	.334	.477	.408
	USGAN$_G$	.296$^{\blacktriangle}$	.279$^{\blacktriangle}$	.810$^{\blacktriangle}$	.508$^{\blacktriangle}$	.398$^{\blacktriangle}$	.562$^{\blacktriangle}$	.479$^{\blacktriangle}$
	USGAN$_D$	.320$^{\blacktriangle}$	.302$^{\blacktriangle}$	.840$^{\vartriangle}$	.560$^{\blacktriangle}$	.426$^{\blacktriangle}$	.610$^{\blacktriangle}$	.514$^{\blacktriangle}$
2006	TF-IDF	.275	.288	.812	.420	.420	.365	.375
	LDA	.265	.284	.803	.376	.390	.331	.348
	AuthorT	.269	.285	.807	.392	.400	.348	.359
	Model 1	.281	.291	.816	.441	.431	.387	.389
	Model 2	.242	.258	.837	.404	.398	.330	.338
	UESM	.473	.410	.882	.727	.667	.669	.662
	USGAN$_G$	.487$^{\blacktriangle}$	.479$^{\blacktriangle}$	.931$^{\blacktriangle}$	.767$^{\blacktriangle}$	.708$^{\blacktriangle}$	.700$^{\blacktriangle}$	.693$^{\blacktriangle}$
	USGAN$_D$	.505$^{\blacktriangle}$	.495$^{\blacktriangle}$	.944$^{\vartriangle}$	.800$^{\blacktriangle}$	.727$^{\blacktriangle}$	.723$^{\blacktriangle}$	.725$^{\blacktriangle}$

6.2 Effect of Window Size

RQ2: We vary the size of the window, i.e., the length of m-grams applied in (7) and see how it contributes to the overall retrieval performance. We only report the performance on MAP and P@10 for the TREC 2005 enterprise track as representative, as the plot patterns on other metrics and for the TREC 2006 track are similar. Fig. 1 shows how the expert retrieval performance on MAP and P@10 for the TREC 2005 enterprise track varies against the window size $m = 2^i (0 \leq i \leq 5)$, respectively.

As can be seen from the figures in Fig. 1, there is a significant retrieval performance increase on the metrics with the size of the m-grams from $m = 1$ to $m = 8$ on our models USGAN$_D$, USGAN$_G$, and the baseline UESM, which underlines the importance of the size of m-grams applied in the models. The increase on the metrics implies that the retrieval performance of our models and the baseline model UESM achieved is not solely due to initialization with pre-trained representations of candidate experts and words, but that the models efficiently learn the representations tailored to the problem domain. As the window size of the m-grams increases beyond $m = 8$ the performance of our models, USGAN$_D$



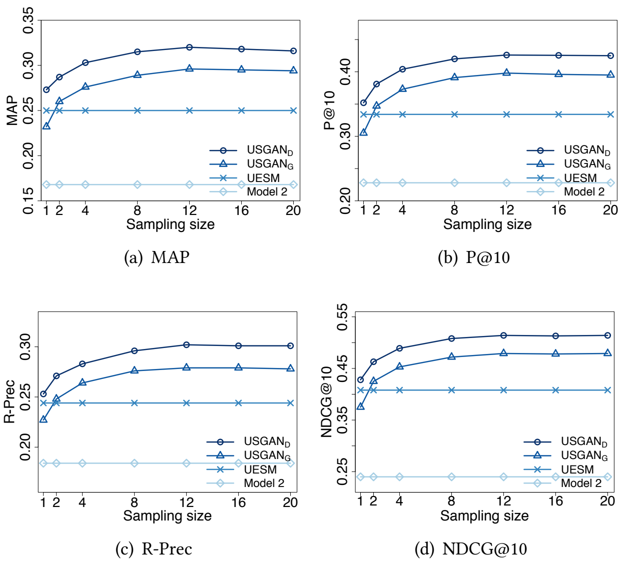
(a) MAP

(b) P@10

(c) R-Prec

(d) NDCG@10

Figure 1: MAP, P@10, R-Prec and NDCG@10 performance of our USGAN retrieval models and the baseline models UESM and Model 2 on the TREC 2005 enterprise track with different size of m-grams, respectively.



(a) MAP

(b) P@10

(c) R-Prec

(d) NDCG@10

Figure 2: MAP, P@10, R-Prec and NDCG@10 performance of our USGAN retrieval models and the baseline models UESM and Model 2 on the TREC 2005 enterprise track with different sampling size H, respectively.

and USGAN$_G$, and the baseline model UESM seems to be level-off. These demonstrate one merit of our USGAN expert retrieval models: they are not sensitive to the window size of m-grams when m is large enough. Other retrieval models, e.g., Model 2, do not consider m-grams for retrieval and thus their performance is consistent over different scales of m-grams. In the next subsection, we report the experimental results with different sampling sizes.

6.3 Effect of Sampling Size

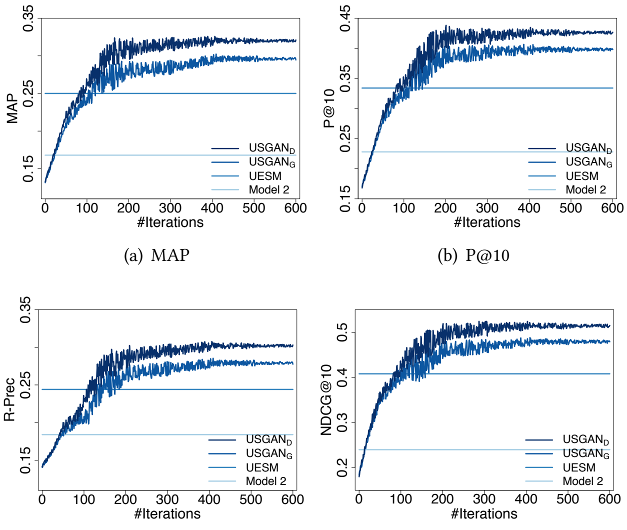
RQ3: To understand the effect of the sampling size, i.e., the parameter H in (10), on the performance of our proposed retrieval model, we vary the size of the samplings H during the REINFORCE learning for updating the parameter $\boldsymbol{\theta}$ in USGAN. In the experiments, as our goal is to yield better expert retrieval performance and to make the iterations converge as fast as possible, for each iteration during the unsupervised training, we choose top-H candidates as the samplings for updating $\boldsymbol{\theta}$.

Fig. 2 shows how the expert retrieval performance varies against the size of the samplings from $H = 1$ to $H = 20$. According to the figures, USGAN$_G$ with the size of samplings being 1 performs a little poorly compared to the baseline UESM that does not need to sample candidates for optimization. This is because one sample is not able to provide enough information for updating the parameter $\boldsymbol{\theta}$ during the REINFORCE learning in our USGAN. USGAN$_D$ with the size of samplings being 1 still can outperforms UESM, and this is because USGAN$_D$ can directly update the parameter $\boldsymbol{\phi}$ by stochastic gradient decent without sampling candidates. There is an obvious performance ascent when tuning the size of samplings from 1 to the optimal value 12, which illustrates the fact that utilizing more top-H samplings for updating the parameter $\boldsymbol{\theta}$ does help to improve the performance when $H \leq 12$. The performance reaches a plato after the optimal number of the samplings $H = 12$. These findings show another merit of our USGAN model: we do not need to sample too many discrete scores, i.e., the top-H candidate experts, at each iteration for the REINFORCE learning, while the model can still perform well. Once again, in Fig. 2 we observe that the retrieval performance of USGAN$_D$ outperforms that of USGAN$_G$ on all the different sampling sizes. This is, again, because the parameter updating strategy of USGAN$_D$ is different from that of USGAN$_G$: USGAN$_D$ directly optimize its parameter $\boldsymbol{\phi}$ with its objective function, whereas USGAN$_G$ needs to leverage REINFORCE learning to update its parameter $\boldsymbol{\theta}$. The baseline model UESM that works with a common neural network does not need to sample candidate experts for parameter updating.

6.4 Impact of Numbers of Iterations

RQ4: To figure out the impact of the number of iterations on the retrieval performance, we plot the expert retrieval performance curves of our USGAN$_D$ and USGAN$_G$ models with different number of iterations from 0 to 600 on representative evaluation metrics MAP, P@10, R-Prec and NDCG@10 in Fig. 3.

As is shown in Fig. 3 before we start the iterations, i.e., the number of iterations being 0, we can still get not too bad retrieval performance that is somewhat a litter worse than that of the baseline language model, i.e., Model 2. This is because we pre-train our USGAN$_D$ and USGAN$_G$ with all documents in $\mathcal{D}$ by (7) (step 2 in Algorithm 1) before we perform the iterations. The performance of both USGAN$_D$ and USGAN$_G$ gradually increase with the iterations increase from 0 to 20. In addition, we can observe that after about 50 iterations, USGAN$_D$ consistently outperforms USGAN$_G$ and such performance gap between them seems to increase with more iterations. These findings again confirm the fact that our updating strategies, i.e., the REINFORCE updating strategy for obtaining the optimal parameter $\boldsymbol{\theta}$ in USGAN$_G$ applied in USGAN and the



(a) MAP

(b) P@10

(c) R-Prec

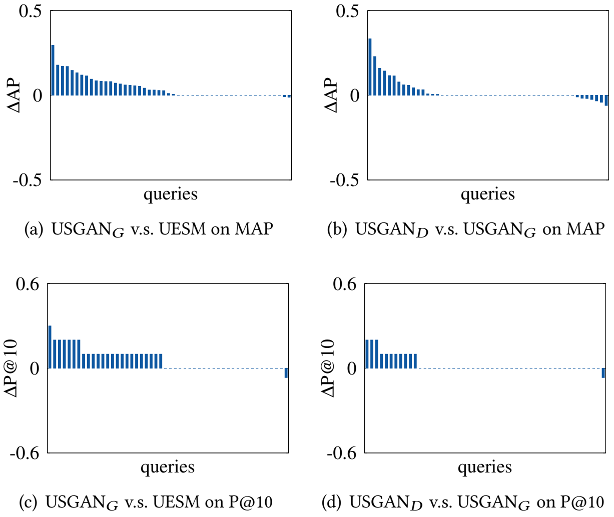
(d) NDCG@10

Figure 3: MAP, P@10, R-Prec and NDCG@10 expert retrieval performance on the TREC 2005 enterprise track with the number of iterations from 0 to 600, respectively. Figures should be viewed in color.

stochastic gradient descent updating strategy for obtaining the optimal parameter $\boldsymbol{\phi}$, are effective. The performance of both USGAN$_D$ and USGAN$_G$ dramatically changes up and down between about 120 iteration and 200 iterations. The performance of both models reaches a plato after about 400 iterations, which illustrates another merit of the proposed USGAN: it is not sensitive to the number of iterations once we have performed enough iterations.

6.5 Query-Level Analysis

RQ5: Last, we take an in-depth view of the improvements of USGAN$_G$ over the best baseline (UESM) and USGAN$_D$ over USGAN$_G$ on a per query basis, respectively. We only report the result on MAP and P@10 for the TREC 2005 enterprise track, as the results on other metrics have similar performance patterns. Fig. 4(a) and Fig. 4(b) show the per query AP performance differences between USGAN$_G$ and the best baseline UESM, and between our discriminative model USGAN$_D$ and our generative model USGAN$_G$, respectively; whereas Fig. 4(c) and Fig. 4(d) show the per query P@10 performance differences, respectively. As can be seen in Fig. 4(a), the number of queries on which USGAN$_G$ outperforms UESM is absolutely more than the number of queries on which UESM outperforms USGAN$_G$. Similar patterns can be found in Fig. 4(b) and Fig. 4(d), where the number of queries on which USGAN$_D$ outperforms USGAN$_G$ is more than that the number of queries on which USGAN$_G$ outperforms USGAN$_D$ on the metrics, respectively. These findings further support the conclusion that our unsupervised generative adversarial networks, USGAN$_D$ and USGAN$_G$, are able to retrieve more relevant experts for a given query compared to that of the best baseline model, UESM, and our discriminative model, USGAN$_D$, works better than our generative model, USGAN$_G$.



(a) USGAN$_G$ v.s. UESM on MAP

(b) USGAN$_D$ v.s. USGAN$_G$ on MAP

(c) USGAN$_G$ v.s. UESM on P@10

(d) USGAN$_D$ v.s. USGAN$_G$ on P@10

Figure 4: Per-query MAP retrieval performance differences on the 2005 track (a) between USGAN$_G$ and UESM on MAP, (b) between USGAN$_D$ and USGAN$_G$ on MAP, (c) between USGAN$_G$ and UESM on P@10, and b) between USGAN$_D$ and USGAN$_G$ on P@10, respectively. A bar above the line $y = 0$ indicates the former model outperforms the latter model, while the opposite is true for bars below $y = 0$, respectively.

7 CONCLUSIONS

We have studied the problem of expert retrieval in enterprise corpora. We have proposed an unsupervised semantic minimax two-player game, i.e., unsupervised semantic generative adversarial networks (USGAN), for the expert retrieval. Our USGAN consists of a discriminative model and a generative model, and semantically represent both candidate experts and words to alleviate the mismatch problems among queries, documents and candidate experts. The unsupervised training signal for our semantic generative network G is provided by our semantic discriminator network D that is trained to distinguish between candidate experts from G and other potential better candidates. Competition in the game drives both models to improve their exert retrieval performance. Unlike most previous generative adversarial networks-based algorithms that require a large amount of labelled data, we train our USGAN in an unsupervised way to obtain the semantic representations of both experts and words, where we consider the candidate-document associations for the optimization during the training. To obtain optimal parameters in our generative model, we apply REINFORCE policy gradient during the iterations. Importantly, our USGAN goes beyond tackling the expert retrieval task in unsupervised way, as it may provide insight and inspiration on how to extend supervised GAN-based models in unsupervised ways to tackle other tasks where labelled data is missing. We found that our proposed USGAN outperforms all the state-of-the-art unsupervised expert retrieval algorithms, including both semantic or non-semantic ones.

Looking forward, there are many unexplored avenues. For instance, how to extend the supervised standard GAN and its variants via unsupervised ways to tackle other tasks where no manually labelled data is available? Can we use implicit signals such as clicks of documents into our USGAN to improve performance of document retrieval? Can we apply our USGAN for expert profiling [6, 8, 30, 41] in enterprise corpora?

REFERENCES

[1] Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862* (2017).

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875* (2017).

[3] Peter Bailey, Arjen P De Vries, Nick Craswell, and Ian Soboroff. 2007. Overview of the TREC 2007 Enterprise Track. In *Proceedings of Text REtrieval Conference.* 1–7.

[4] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2006. Formal Models for Expert Finding in Enterprise Corpora. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* 43–50.

[5] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2009. A language modeling framework for expert finding. *Information Processing & Management* 45, 1 (2009), 1–19.

[6] Krisztian Balog, Toine Bogers, Leif Azzopardi, Maarten de Rijke, and Antal van den Bosch. 2007. Broad Expertise Retrieval in Sparse Data Environments. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* 551–558.

[7] Krisztian Balog, Maarten de Rijke, and Wouter Weerkamp. 2008. Bloggers As Experts: Feed Distillation Using Expert Retrieval Models. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* 753–754.

[8] Krisztian Balog, Yi Fang, Maarten de Rijke, Pavel Serdyukov, and Luo Si. 2012. Expertise Retrieval. *Found. Trends Inf. Retr.* 6 (2012), 127–256.

[9] Krisztian Balog, Ian Soboroff, Paul Thomas, Peter Bailey, Nick Craswell, and Arjen P. de Vries. 2008. Overview of the TREC 2008 Enterprise Track. In *Proceedings of Text REtrieval Conference.* 1–12.

[10] Richard Berendsen, Maarten Rijke, Krisztian Balog, Toine Bogers, and Antal Bosch. 2013. On the assessment of expertise profiles. *Journal of the Association for Information Science and Technology* 64, 10 (2013), 2024–2044.

[11] David Berthelot, Tom Schumm, and Luke Metz. 2017. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717* (2017).

[12] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

[13] Zheqian Chen, Ben Gao, Huimin Zhang, Zhou Zhao, Haifeng Liu, and Deng Cai. 2017. User Personalized Satisfaction Prediction via Multiple Instance Deep Learning. In *Proceedings of the International World Wide Web Conference.* 907–915.

[14] Nick Craswell, Arjen P. de Vries, and Ian Soboroff. 2005. Overview of the TREC 2005 Enterprise Track. In *Proceedings of Text REtrieval Conference.* 1–7.

[15] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2015. *Search engines: Information retrieval in practice.* Addison-Wesley Reading.

[16] Gianluca Demartini, Julien Gaugaz, and Wolfgang Nejdl. 2009. A Vector Space Model for Ranking Entities and Its Application to Expert Search.. In *Proceedings of the European Conference on Information Retrieval.* 189–201.

[17] Hui Fang and ChengXiang Zhai. 2007. Probabilistic models for expert finding. In *Proceedings of the European Conference on Information Retrieval.* 418–430.

[18] Yi Fang, Luo Si, and Aditya P. Mathur. 2010. Discriminative Models of Integrating Document Evidence and Document-Candidate Associations for Expert Search. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* 683–690.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the Conference on Neural Information Processing Systems.* 2672–2680.

[21] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the ACM International on Information and Knowledge Management.* 55–64.

[22] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* 50–57.

[23] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the ACM International on Information and Knowledge Management.* 2333–2338.

[24] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. Position-Aware Representations for Relevance Matching in Neural Information Retrieval. In *Proceedings of the International World Wide Web Conference.* 799–800.

[25] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations.* 1–13.

[26] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.

[27] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image Generation from Scene Graphs. *arXiv preprint arXiv:1804.01622* (2018).

[28] Viveka Kulharia, Arnab Ghosh, Amitabha Mukerjee, Vinay Namboodiri, and Mohit Bansal. 2017. Contextual RNN-GANs for abstract reasoning diagram generation. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 2852–2858.

[29] Hang Li and Jun Xu. 2014. Semantic Matching in Search. *Found. Trends Inf. Retr.* 7, 5 (2014), 343–469.

[30] Shangsong Liang. 2019. Collaborative, Dynamic and Diversified User Profiling. In *AAAI.*

[31] Shangsong Liang, Fei Cai, Zhaochun Ren, and Maarten de Rijke. 2016. Efficient Structured Learning for Personalized Diversification. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 2958–2973.

[32] Shangsong Liang and Maarten de Rijke. 2013. Finding knowledgeable groups in enterprise corpora. In *SIGIR.* 1005–1008.

[33] Shangsong Liang and Maarten de Rijke. 2015. Burst-aware data fusion for microblog search. *Information Processing & Management* (2015), 89–113.

[34] Shangsong Liang and Maarten de Rijke. 2016. Formal language models for finding groups of experts. *Information Processing & Management* (2016), 529–549.

[35] Shangsong Liang, Maarten de Rijke, and Manos Tsagkias. 2013. Late Data Fusion for Microblog Search. In *Proceedings of the European Conference on Information Retrieval.* 743–746.

[36] S. Liang, Z. Ren, and M. de Rijke. 2014. Fusion helps diversification. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[37] Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. 2014. Personalized Search Result Diversification via Structured Learning. In *KDD.* 751–760.

[38] Shangsong Liang, Zhaochun Ren, Yukun Zhao, Jun Ma, Emine Yilmaz, and Maarten De Rijke. 2017. Inferring Dynamic User Interests in Streams of Short Texts for User Clustering. *ACM Trans. Inf. Syst.* 36, 1 (July 2017), 10:1–10:37.

[39] Shangsong Liang, Emine Yilmaz, and Evangelos Kanoulas. 2016. Dynamic Clustering of Streaming Short Documents. In *KDD.* 995–1004.

[40] Shangsong Liang, Emine Yilmaz, and Evangelos Kanoulas. 2019. Collaboratively Tracking Interests for User Clustering in Streams of Short Texts. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2019), 257–272.

[41] Shangsong Liang, Xiangliang Zhang, Zhaochun Ren, and Evangelos Kanoulas. 2018. Dynamic Embeddings for User Profiling in Twitter. In *KDD.* ACM, 1764–1773.

[42] Francesco Locatello, Damien Vincent, Ilya Tolstikhin, Gunnar Rätsch, Sylvain Gelly, and Bernhard Schölkopf. 2018. Clustering Meets Implicit Generative Models. *arXiv preprint arXiv:1804.11130* (2018).

[43] Genevieve B Orr and Klaus-Robert Müller. 2012. *Neural networks: tricks of the trade (second edition).* Springer.

[44] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing.* 1532–1543.

[45] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *UAI.* 487–494.

[46] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Proceedings of the Conference on Neural Information Processing Systems.* 2234–2242.

[47] Othman Sbai, Mohamed Elhoseiny, Antoine Bordes, Yann LeCun, and Camille Couprie. 2018. DeSIGN: Design Inspiration from Generative Networks. *arXiv preprint arXiv:1804.00921* (2018).

[48] Ian Soboroff, Arjen P de Vries, and Nick Craswell. 2006. Overview of the TREC 2006 Enterprise Track. In *TREC'06.* 1–20.

[49] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the Conference on Neural Information Processing Systems.* 1057–1063.

[50] Christophe Van Gysel, Maarten de Rijke, and Marcel Worring. 2016. Unsupervised, efficient and semantic expertise retrieval. In *Proceedings of the International World Wide Web Conference.* ACM, 1069–1079.

[51] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* 515–524.

[52] Wei Wei, Gao Cong, Chuanyan Miao, Feida Zhu, and Guohui Li. 2016. Learning to Find Topic Experts in Twitter via Different Relations. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2016), 1764–1778.

[53] Xing Wei and W. Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the International ACM SIGIR Conference on Research*

and Development in Information Retrieval. 178–185.

[54] Ronald J. Williams. 1992. *Simple statistical gradient-following algorithms for connectionist reinforcement learning*. Springer.

[55] Zhe Xu and Jay Ramanathan. 2016. Thread-based probabilistic models for expert finding in enterprise Microblogs. *Expert Systems with Applications* (2016), 286–297.

[56] Xing Yi and James Allan. 2009. A Comparative Study of Utilizing Topic Models for Information Retrieval. In *Proceedings of the European Conference on Information Retrieval*. 29–41.

[57] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2852–2858.

[58] Junbo Zhao, Michael Mathieu, and Yann LeCun. 2017. Energy-based generative adversarial network. In *Proceedings of the International Conference on Learning Representations*. 1–10.