Towards Better Text Understanding and Retrieval through Kernel Entity Salience Modeling

Chenyan Xiong
Carnegie Mellon University
cx@cs.cmu.edu

Zhengzhong Liu
Carnegie Mellon University
liu@cs.cmu.edu

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu

Tie-Yan Liu
Microsoft Research
tie-yan.liu@microsoft.com

ABSTRACT

This paper presents a Kernel Entity Salience Model (KESM) that improves text understanding and retrieval by better estimating entity salience (importance) in documents. KESM represents entities by knowledge enriched distributed representations, models the interactions between entities and words by kernels, and combines the kernel scores to estimate entity salience. The whole model is learned end-to-end using entity salience labels. The salience model also improves ad hoc search accuracy, providing effective ranking features by modeling the salience of query entities in candidate documents. Our experiments on two entity salience corpora and two TREC ad hoc search datasets demonstrate the effectiveness of KESM over frequency-based and feature-based methods. We also provide examples showing how KESM conveys its text understanding ability learned from entity salience to search.

建模词和实体之间的交互，从而评估实体的重要度.

评估

KEYWORDS

Text Understanding, Entity Salience, Entity-Oriented Search

1 INTRODUCTION

Natural language understanding has been a long desired goal in information retrieval. In search engines, the process of text understanding begins with the representations of query and documents. The representations can be bag-of-words, the set of words in the text, or bag-of-entities, which uses automatically linked entity annotations to represent texts [10, 20, 25, 29].

With the representations, the next step is to estimate the term (word or entity) importance in text, which is also called term *salience* estimation [8, 9]. The ability to know which terms are salient (important and central) to the meaning of texts is crucial to many text-related tasks. In ad hoc search, the document ranking is often determined by the salience of query terms in them, which is typically estimated by combining frequency-based signals such as term frequency and inverse document frequency [5].

信息检索:
①.文本理解
②term重要性评估

Effective as it is, frequency is not equal to salience. For example, a Wikipedia article about an entity may not repeat the entity the most frequently; a person's homepage may only mention her name once; a frequently mentioned term may be a stopword. In word-based retrieval, many approaches have been developed to better estimate term importance [3]. However, in entity-based representations [20, 26, 29], while entities convey richer semantics [1], entity salience estimation is a rather immature task [8, 9] and its effectiveness in search has not yet been explored.

在基本实体的表示中，实体的重要性评估方法还不成熟.

This paper focuses on improving text understanding and retrieval by better estimating *entity salience* in documents. We present a Kernel Entity Salience Model (KESM) that estimates entity salience end-to-end using neural networks. Given annotated entities in a document, KESM represents them using Knowledge Enriched Embeddings and models the interactions between entities and words using a Kernel Interaction Model [27]. In the entity salience task [9], the kernel scores from the interaction model are combined by KESM to estimate entity salience, and the whole model, including the Knowledge Enriched Embeddings and Kernel Interaction Model, is learned end-to-end using a large number of salience labels.

基本思路

KESM also improves ad hoc search by modeling the salience of query entities in candidate documents. Given a query-document pair and their entities, KESM uses its kernels to model the interactions of *query entities* with the entities and words in the document. It then merges the kernel scores to ranking features and combines these features to rank documents. In ad hoc search, KESM can either be trained end-to-end when sufficient ranking labels are available, or be first pre-trained on the salience task and then adapted to search as a salience ranking feature extractor.

建模query entity和document word之间的交互，从而提高搜索性能.

Our experiments on a news corpus [9] and a scientific proceeding corpus [29] demonstrate KESM's effectiveness in the entity salience task. It outperforms previous frequency-based and feature-based models by large margins, while requires much less linguistic pre-processing than the feature-based model. Our analyses find that KESM has a better balance on popular (head) entities and rare (tail) entities when predicting salience. In contrast, frequency-based or feature-based methods are heavily biased towards the most popular entities—less attractive to users as they are more expected. Also, KESM is less sensitive to document length while frequency-based methods are not as effective on shorter documents.

Our experiments on TREC Web Track search tasks show that KESM's text understanding ability in estimating entity salience also improves search accuracy. The salience ranking features from KESM, pre-trained on the news corpus, outperform both word-based and

SIGIR '18, Ann Arbor, MI, USA
© 2018 ACM. 978-1-4503-5657-2/18/07…$15.00
DOI: 10.1145/3209978.3209982

entity-based features in learning to rank, despite various differences in the salience and search tasks. Our case studies find interesting examples showing that KESM favors documents centering on query entities over those merely mentioning them. We find it encouraging that the fine-grained text understanding ability of KESM—the ability to model the consistency and interactions between entities and words in texts—is indeed valuable to ad hoc search.

The next section discusses related work. Section 3 describes the Kernel Entity Salience Model and its application to entity salience estimation. Section 4 discusses its application to ad hoc search. Experimental methodology and results for entity salience are presented in Sections 5 and Section 6. Those for ad hoc search are in Sections 7 and Section 8. Section 9 concludes.

2 RELATED WORK

Representing and understanding texts is a key challenge in information retrieval. The standard approaches in modern information retrieval represent a text by a bag-of-words; they model term importance using frequency-based signals such as term frequency (TF), inverse document frequency (IDF), and document length [5]. The bag-of-words representation and frequency-based signals are the backbone of modern information retrieval and have been used by many unsupervised and supervised retrieval models [5, 14].

Nevertheless, bag-of-words and frequency-based statistics only provide shallow text understanding. One way to improve the text understanding is to use more meaningful language units than words in text representations. These approaches include the first generation of search engines that were based on controlled vocabularies [5] and also the recent entity-oriented search systems which utilize knowledge graphs in search [7, 15, 20, 24, 29]. In these approaches, texts are often represented by entities, which introduce information from knowledge graphs to search systems.

In both word-based and entity-based text representations, frequency signals such as TF and IDF provide good approximations for the importance or salience of terms (words or entities) in the query or documents. However, solely relying on frequency signals limits the search engine's text understanding capability; many approaches have been developed to improve *term importance estimation*.

In the word space, the query term weighting research focuses on modeling the importance of words or phrases in the query. For example, Bendersky et al. use a supervised model to combine the signals from Wikipedia, search log, and external collections to better estimate term importance in verbose queries [2]; Zhao and Callan predict the necessity of query terms using evidence from pseudo relevance feedback [30]; word embeddings have also been used as features in supervised query term importance prediction [31]. These methods in general leverage extra signals to model how important a term is to capture search intents. They can improve the performance of retrieval models compared to frequency-based term weighting.

The word importance in documents can also be estimated by graph-based approaches [3, 18, 21]. Instead of using isolated words, the graph-based approaches connect words by co-occurrence or proximity. Then graph ranking algorithms, for example, PageRank, are used to estimate term importance in a document. The graph ranking scores reflect the centrality and connectivity of words and are able to improve standard retrieval models [3, 21].

In the entity space, modeling term importance is even more crucial. Unlike word-based representations, the entity-based representations are often automatically constructed and inevitably include noises. The noisy query entities have been a major bottleneck for entity-oriented search and often required manual cleaning [7, 10, 15]. Along this line, a series of approaches have been developed to model the importance of entities in a query, for example, latent-space learning to rank [23] and hierarchical ranking models [26]. These approaches learn the importance of query entities and the ranking of documents jointly using ranking labels. The features used to describe the entity importance include IR-style features [23] and NLP-style features from entity linking [26].

Nevertheless, previous research on modeling entity salience mainly focused on query representations, while the entities in document representations are still weighted by frequencies, i.e. in the bag-of-entities model [26, 29]. Recently, Dunietz and Gillick [9] proposed the entity salience task using the New York Times corpus [22]; they consider the entities that are annotated in the expert-written summary to be salient to the article, enabling them to automatically construct millions of training data. Dojchinovski et al. constructed a deeper study and found that crowdsource workers consider entity salience an intuitive task [8]. Both of them demonstrated that the frequency of an entity is not equal to its salience; a supervised model with linguistic and semantic features is able to outperform frequency significantly, though mixed findings have been found with graph-based methods such as PageRank.

3 KERNEL ENTITY SALIENCE MODEL

This section presents our Kernel Entity Salience Model (KESM). Compared to the feature-based salience models [8, 9], KESM uses neural networks to learn the representation of entities and their interactions for salience estimation.

The rest of this section first describes the overall architecture of KESM and then how it is applied to the entity salience task.

3.1 Model Architecture

As shown in Figure 1, KESM includes two main components: the *Knowledge Enriched Embedding* (Figure 1a) and the *Kernel Interaction Model* (Figure 1b).

KESM包含
2个模块

Knowledge Enriched Embedding (KEE) encodes each entity e into its distributed representation $\vec{v}_e$. It is achieved by first using an embedding layer that maps the entity to an embedding:

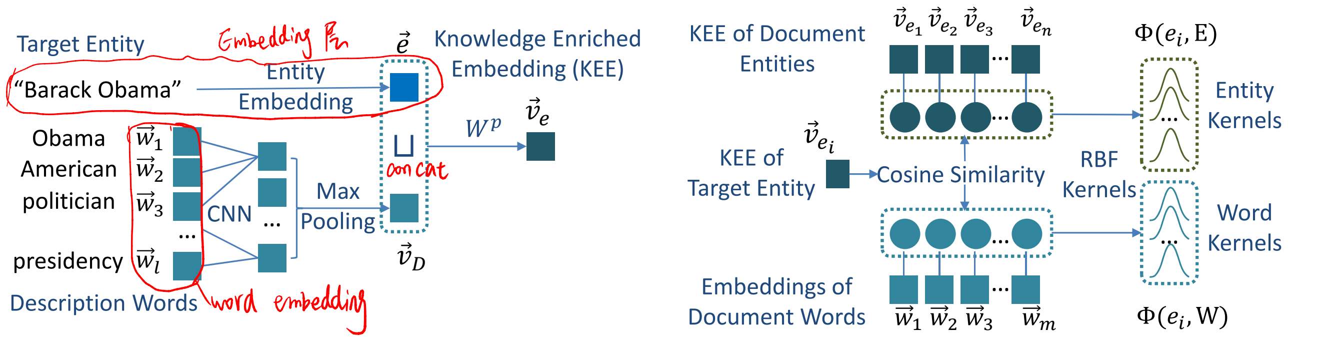
$$\boxed{e \xrightarrow{V} \vec{e}. \qquad \text{Entity Embedding} \quad \text{layer}}$$

V is the parameters of the embedding layer to be learned.

An advantage of entities is that they are associated with external semantics in the knowledge graph, for example, synonyms, descriptions, types, and relations. Instead of only using $\vec{e}$, KEE enriches the entity representation with its description, for example, the first paragraph of its Wikipedia page.

(用知识库扩充实体的表示)

Specifically, given the description $\mathbb{D}$ of the entity e, KEE uses a Convolutional Neural Network (CNN) to compose the words in $\mathbb{D}$:



(a) Knowledge Enriched Embedding (KEE)

(b) Kernel Interaction Model (KIM)

Figure 1: KESM Architecture. (a): Entities are represented using embeddings enriched by their descriptions. (b): The salience of an entity in a document is estimated by kernels that model its interactions with entities and words in the document. Squares are continuous vectors (embeddings) and circles are scalars (cosine similarities).

$\{w_1, ..., w_p, ..., w_l\}$, into one embedding:

$$\boxed{\begin{aligned} w_p &\xrightarrow{V} \vec{w}_p, && \text{Word Embedding} \\ C_p &= W^c \cdot \vec{w}_{p:p+h}, && \text{CNN Filter} \\ \vec{v}_{\mathbb{D}} &= \max(C_1, ..., C_p, ..., C_{l-h}). && \text{Description Embedding} \end{aligned}}$$

It embeds the words into $\vec{w}$ using the embedding layer, composes the word embeddings using CNN filters, and generates the description embeddings $\vec{v}_{\mathbb{D}}$ using max-pooling. W^c and h are the weights and length of the CNN.

$\vec{v}_D$ is then combined with the entity embedding $\vec{e}$ by projection:

$$\boxed{\vec{v}_e = W^p \cdot \underbrace{(\vec{e} \sqcup \vec{v}_{\mathbb{D}})}_{concat}. \qquad \text{KEE Embedding}}$$

⊔ is the concatenation operator and W^p is the projection weights. $\vec{v}_e$ is the KEE vector for e. It incorporates the external information from the knowledge graph and is to be learned as part of KESM.

Kernel Interaction Model (KIM) models the interactions of a target entity with entities and words in the document using their distributed representations.

Given a document d, its annotated entities $\mathbb{E} = \{e_1, ...e_i..., e_n\}$, and its words $\mathbb{W} = \{w_1, ...w_j..., w_m\}$, KIM models the interactions of a target entity e_i with $\mathbb{E}$ and $\mathbb{W}$ using kernels [6, 27]:

$$\boxed{KIM(e_i, d) = \Phi(e_i, \mathbb{E}) \underset{concat}{\sqcup} \Phi(e_i, \mathbb{W}).} \tag{1}$$

The entity kernels $\Phi(e_i, \mathbb{E})$ model the interaction between e_i and document entities $\mathbb{E}$:

第k个径向基核函数

$$\Phi(e_i, \mathbb{E}) = \{\phi_1(e_i, \mathbb{E}), ...\phi_k(e_i, \mathbb{E})..., \phi_K(e_i, \mathbb{E})\}, \tag{2}$$

建模 e_i 与其他实体的交互.

$$\phi_k(e_i, \mathbb{E}) = \sum_{e_j \in \mathbb{E}} \exp\left(-\frac{\left(cos(\vec{v}_{e_i}, \vec{v}_{e_j}) - \mu_k\right)^2}{2\sigma_k^2}\right). \tag{3}$$

$\vec{v}_{e_i}$ and $\vec{v}_{e_j}$ are the KEE embeddings of e_i and e_j. $\phi_k(e_i, \mathbb{E})$ is the k-th RBF kernel with mean μ_k and variance σ_k^2. If $(\mu_k = 1, \sigma_k \to \infty)$, ϕ_k counts the entity frequency. Otherwise, it models the interactions between the target entity e_i and other entities in the KEE representation space. One view of kernels is that they count the number of entities whose similarities with e_i are in its region (μ_k, σ_k^2); the other view is that the kernel scores are the votes from other entities in a certain neighborhood (kernel region) of the current entity.

对径向基核函数的解释

当 $\mu_k=1$, $\sigma_k=\infty$ 时, ϕ_k 表示 e_i 在 E 中出现的次数.

否则, ϕ_k 表示与 e_i 的相似度在 (μ_k, σ_k^2) 范围的实体数.

Similarly, the word kernels $\Phi(e_i, \mathbb{W})$ model the interactions between e_i and document words $\mathbb{W}$:

$$\Phi(e_i, \mathbb{W}) = \{\phi_1(e_i, \mathbb{W}), ...\phi_k(e_i, \mathbb{W})..., \phi_K(e_i, \mathbb{W})\}, \tag{4}$$

$$\phi_k(e_i, \mathbb{W}) = \sum_{w_j \in \mathbb{W}} \exp\left(-\frac{\left(cos(\vec{v}_{e_i}, \vec{w}_j) - \mu_k\right)^2}{2\sigma_k^2}\right). \tag{5}$$

$\vec{w}_j$ is the word embedding of w_j, mapped by the same embedding parameters (V). The word kernels $\phi_k(e_i, \mathbb{W})$ model the interactions between e_i and document words, gathering 'votes' from words for e_i in the corresponding kernel regions.

For each entity e_i, KEE encodes it to $\vec{v}_{e_i}$ and KIM models its interactions with entities and words in the document. The kernel scores $KIM(e_i, d)$ include signals from three sources: the description of the entity in the knowledge graph, its interactions with the document entities, and its interactions with the document words. The utilization of these kernel scores depends on the specific task: entity salience estimation (Section 3.2) or document ranking (Section 4).

3.2 Entity Salience Estimation

The application of KESM in the entity salience task is simple. Combining the KIM kernel scores gives the salience score of the corresponding entity:

$$\underbrace{f(e_i, d)}_{\text{表示}e_i\text{在}d\text{中的重要度}} = W^s \cdot KIM(e_i, d) + b^s. \tag{6}$$

$f(e_i, d)$ is the salience score of e_i in d. W^s and b^s are parameters for salience estimation.

Learning: The entity salience training data are labels about document-entity pairs that indicate whether the entity is salient to the document. The salience label of entity e_i to document d is:

$$y(e_i, d) = \begin{cases} +1, & \text{if } e_i \text{ is a salient entity in } d; \\ -1, & \text{otherwise.} \end{cases}$$

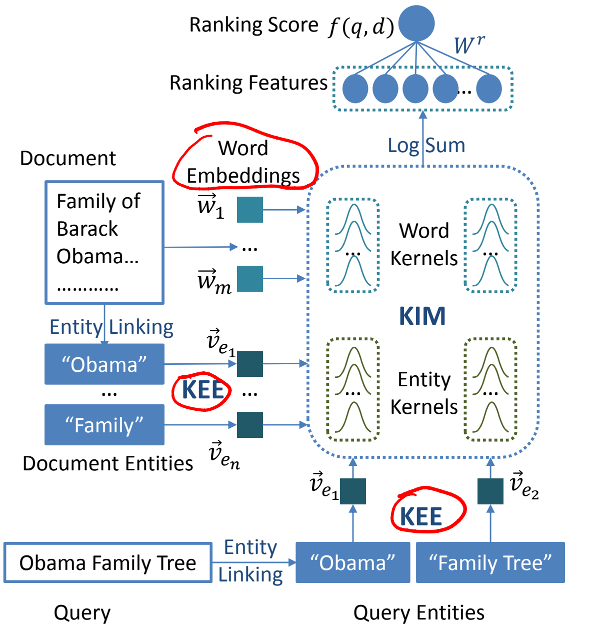


Figure 2: Ranking with KESM. KEE embeds the entities. KIM calculates the kernel scores of query entities VS. document entities and words. The kernel scores are combined to ranking features and then to the ranking score.

We use pairwise learning to rank [14] to train KESM:

$$\sum_{e^+, e^- \in d} \max(0, 1 - f(e^+, d) + f(e^-, d)), \tag{7}$$

$$\text{w.r.t. } y(e^+, d) = +1 \;\&\; y(e^-, d) = -1.$$

端到端
训练 KEE+KIM

The loss function enforces KESM to rank the salient entities e^+ ahead of the non-salient ones e^- within the same document.

In the entity salience task, KESM is trained end-to-end by back-propagation. During training, the gradients from the labels are first propagated to the Kernel Interaction Model (KIM) and then the Knowledge Enriched Embedding (KEE). KESM updates the kernel weights; KIM converts the gradients from kernels to 'expectations' on the distributed representations—how the entities and words should be allocated in the space to better reflect salience; KEE updates its embeddings and parameters according to these 'expectations'. The knowledge learned from the training labels is encoded and stored in the model parameters, mainly the embeddings [27].

4 RANKING WITH ENTITY SALIENCE

This section presents the application of KESM in ad hoc search.

Ranking: Knowing which entities are salient in a document indicates a deeper text understanding ability [8, 9]. The improved text understanding should also improve search accuracy: the salience of query entities in a document reflects how focused the document is on the query, which is a strong indicator of relevancy. For example, a web page that exclusively discusses Barack Obama's family is more relevant to the query "Obama Family Tree" than those that just mention his family members.

Table 1: Datasets used in the entity salience task. New York Times are news articles and salient entities are those in the expert-written news summaries. Semantic Scholar are paper abstracts and salient entities are those in the titles.

	New York Times			Semantic Scholar		
	Train	Dev	Test	Train	Dev	Test
# of Documents	526k	64k	64k	800k	100k	100k
Entities Per Doc	198	197	198	66	66	66
Salience Per Doc	27.8	27.8	28.2	7.3	7.3	7.3
Unique Word	609k	278k	281k	921k	300k	301k
Unique Entity	622k	319k	317k	331k	162k	162k

The ranking process of KESM following this intuition is illustrated in Figure 2. It first calculates the kernel scores of the query entities in the document using KEE and KIM. Then it merges the kernel scores from multiple query entities to ranking features and uses a ranking model to combine these features.

Specifically, given query q, query entities $\mathbb{E}^q$, candidate document d, document entities $\mathbb{E}^d$, and document words $\mathbb{W}^d$, the ranking score is calculated as:

$$f(q,d) = W^r \cdot \Psi(q,d), \tag{8}$$

$$\Psi(q,d) = \sum_{e_i \in \mathbb{E}^q} \log \left(\frac{KIM(e_i, d)}{|\mathbb{E}^d|} \right). \tag{9}$$

用于标准化(使不同的文档间有可比性)

$KIM(e_i, d)$ are the kernel scores of the query entity e_i in document d, calculated by the KIM and KEE modules described in last section. $|\mathbb{E}^d|$ is the number of entities in d. W^r is the ranking parameters and $\Psi(q,d)$ are the salience ranking features.

Several adaptations have been made to apply KESM in search. First, Equation (9) normalizes the kernel scores by the number of entities in the document ($|\mathbb{E}^d|$), making them more comparable across *different* documents. In the entity salience task, this is not required because the goal is to distinguish salient entities from non-salient ones in the *same* document. Second, there can be multiple entities in the query and their kernel scores need to be combined to model query-document relevance. The combination is done by log-sum, following language model approaches [5].

Learning: In the search task, KESM is trained using standard pairwise learning to rank and relevance labels:

$$\sum_{d^+ \in D^+, d^- \in D^-} \max(0, 1 - f(q, d^+) + f(q, d^-)). \tag{10}$$

两种训练方式:
①.端到端训练整个KESM和W^r. ②可以先在实体显著度任务上训练KESM,再在ad-hoc retrieval任务上训练W^r.

D^+ and D^- are the relevant and irrelevant documents. $f(q, d^+)$ and $f(q, d^-)$ are the ranking scores calculated by Equation (8).

There are two ways to train KESM for ad hoc search. First, when sufficient ranking labels are available, for example, in commercial search engines, the whole KESM model can be learned end-to-end by back-propagation from Equation (10). On the other hand, when not enough ranking labels are available for end-to-end learning, the KEE and KIM can be first trained using the labels from the entity salience task. Only the ranking parameters W^r need to be learned from relevance labels. As a result, the knowledge learned from the salience labels is adapted to ad hoc search through the ranking features, which can be used in any learning to rank system.

Table 2: Entity salience features used by the LeToR baseline [9]. The features are extracted via various natural language processing techniques, as listed in the Source column.

Name	Description	Source
Frequency	The frequency of the entity	Entity Linking
First Location	The location of the first sentence that contains the entity	Entity Linking
Head Word Count	The frequency of the entity's first head word in parsing	Dependency Parsing
Is Named Entity	Whether the entity is considered as a named entity	Named Entity Recognition
Coreference Count	The coreference frequency of the entity's mentions	Entity Coreference Resolution
Embedding Vote	Votes from other entities through cosine embedding similarity	Entity Embedding (Skip-gram)

5 EXPERIMENTAL METHODOLOGY FOR ENTITY SALIENCE ESTIMATION

This section presents the experimental methodology for the entity salience task. It mainly follows the setup by Dunietz and Gillick [9] with some revisions to facilitate the applications in search. An additional dataset is also introduced.

Datasets[1] used include *New York Times* and *Semantic Scholar*.

The *New York Times* corpus has been used in previous work [9]. It includes more than half million news articles and expert-written summarizes [22]. Among all entities annotated on a news article, those that also appear in the summary of the article are considered as salient entities; others are not [9].

The *Semantic Scholar* corpus contains one million randomly sampled scientific publications from the index of SemanticScholar.org, the academic search engine from Allen Institute for Artificial Intelligence. The full texts of the papers are not released. Only the abstract and title of the paper content are available. We treat the entities annotated on the abstract as the candidate entities of a paper and those also annotated on the title as salient.

The entity annotations on both corpora are Freebase entities linked by TagMe [11]. *All annotations* are included to ensure coverage, which is important for effective text representations [20, 29].

The statistics of the two corpora are listed in Table 1. The Semantic Scholar corpus has shorter documents (paper abstracts) and a smaller entity vocabulary because its papers are mostly in the computer science and medical science domains.

Baselines: Three baselines from previous research are compared: Frequency, PageRank, and LeToR.

Frequency [9] estimates the salience of an entity by its term frequency. It is a straightforward but effective baseline in many related tasks. IDF is not as effective in entity-based text representations [20, 29], so we used only frequency counts.

PageRank [9] estimates the salience score of an entity using its PageRank score [3]. We conduct a supervised PageRank on a fully connected graph. The nodes are the entities in the document. The edges are the embedding similarities of the connected nodes. The entity embeddings are configured and learned in the same manner as KESM. Similar to previous work [9], PageRank is not as effective in the salience task. The results reported are from the best setup we found: a one-step random walk linearly combined with Frequency.

LeToR [9] is a feature-based learning to rank (entity) model. It is trained using the same pairwise loss with KESM, which we found more effective than the pointwise loss used in prior research [9].

We re-implemented the features used by Dunietz and Gillick [9]. As listed in Table 2, the features are extracted by various linguistic and semantic techniques including entity linking, dependency parsing, named entity recognition, and entity coreference resolution. Besides the standard Frequency count, the Head Word Count considers syntactic signals when counting entities; the Coreference Count considers all mentions that refer to an entity as its appearances when counting frequency.

The entity embeddings are trained on the same corpus using Google's Word2vec toolkit [19]. Entity linking is done by TagMe; all entities are kept [20, 29]. Other linguistic and semantic preprocessing are done by the Stanford CoreNLP toolkit [16].

Compared to Dunietz and Gillick [9], we do not include the headline feature because it uses information from the expert-written summary and does not improve the performance much anyway; we also replace the head-lex feature with Embedding Vote which has similar effectiveness but is more efficient.

Evaluation Metrics: We use the ranking-focused evaluation metrics: Precision@{1, 5} and Recall@{1, 5}. These metrics circumvent the problem of selecting a cutoff threshold for each individual document in classification evaluation metrics [9]. Statistical significances are tested by permutation test with $p < 0.05$.

Implementation Details: The hyper-parameters of KESM are configured following popular choices or previous research. The dimension of entity embeddings, word embeddings, and CNN filters are all set to 128. The kernel pooling layers use the same pre-defined kernels as in previous research [27]: one exact match kernel ($\mu = 1, \sigma = 1e - 3$) and ten soft match kernels equally splitting 仿向量核 the cosine similarity range $[-1, 1]$ ($\mu \in \{-0.9, -0.7, ..., 0.9\}$ and $\sigma = 0.1$). The length of the CNN used to =3 encode entity description is set to 3 which is tri-gram. The entity descriptions are fetched from Freebase. The first 20 words (the gloss sentence) of the description are used. The words or entities that appear less than 2 times in the training corpus are replaced by "Unk_word" or "Unk_entity".

The parameters include the embeddings V, the CNN weights W^c, the projection weights W^p, and the kernel weights W^s, b^s. They are learned end-to-end using Adam optimizer, size 64 mini-batching, and early-stopping on the development split. V is initialized by the skip-gram embeddings of words and entities jointly trained on the training corpora, which takes several hours [26]. With our PyTorch implementation, KESM usually only needs one pass on the training data and converges within several hours on a typical GPU. In comparison, LeToR takes days to extract its features because parsing and coreference are costly.

[1] Available at http://boston.lti.cs.cmu.edu/appendices/SIGIR2018-KESM/

Table 3: Entity salience performances on New York Times and Semantic Scholar. (E), (W), and (K) mark the resources used by KESM: Entity kernels, Word kernels, and Knowledge enrichment. KESM is the full model. Relative performances over LeToR are shown in the percentages. W/T/L are the number of documents a method improves, does not change, and hurts, compared to LeToR. †, ‡, §, and ¶ mark the statistically significant improvements over Frequency†, PageRank‡, LeToR§, and KESM (E)¶.

New York Times									
Method	**Precision@1**		**Precision@5**		**Recall@1**		**Recall@5**		**W/T/L**
Frequency	0.5840	−8.53%	0.4065	−11.82%	0.0781	−11.92%	0.2436	−14.44%	5,622/38,813/19,154
PageRank	0.5845†	−8.46%	0.4069†	−11.73%	0.0782†	−11.80%	0.2440†	−14.31%	5,655/38,841/19,093
LeToR	0.6385	–	0.4610	–	0.0886	–	0.2848	–	–/–/–
KESM (E)	0.6470†‡§	+1.33%	0.4782†‡§	+3.73%	0.0922†‡§	+4.03%	0.3049†‡§	+7.05%	19,778/27,983/15,828
KESM (EK)	0.6528†‡§¶	+2.24%	0.4769†‡§	+3.46%	0.0920†‡§	+3.82%	0.3026†‡§	+6.27%	18,619/29,973/14,997
KESM (EW)	0.6767†‡§¶	+5.98%	0.5018†‡§¶	+8.86%	0.0989†‡§¶	+11.57%	0.3277†‡§¶	+15.08%	22,805/26,436/14,348
KESM	**0.6866**†‡§¶	+7.53%	**0.5080**†‡§¶	+10.21%	**0.1010**†‡§¶	+13.93%	**0.3335**†‡§¶	+17.10%	23,290/26,883/13,416

Semantic Scholar									
Method	**Precision@1**		**Precision@5**		**Recall@1**		**Recall@5**		**W/T/L**
Frequency	0.3944	−9.99%	0.2560	−11.38%	0.1140	−12.23%	0.3462	−13.67%	11,155/64,455/24,390
PageRank	0.3946†	−9.94%	0.2561†	−11.34%	0.1141†	−12.11%	0.3466†	−13.57%	11,200/64,418/24,382
LeToR	0.4382	–	0.2889	–	0.1299	–	0.4010	–	–/–/–
KESM (E)	0.4793†‡§	+9.38%	0.3192†‡§	+10.51%	0.1432†‡§	+10.26%	0.4462†‡§	+11.27%	27,735/56,402/15,863
KESM (EK)	0.4901†‡§¶	+11.84%	0.3161†‡§	+9.43%	0.1492†‡§¶	+14.91%	0.4449†‡§	+10.95%	28,191/54,084/17,725
KESM (EW)	0.5097†‡§¶	+16.31%	0.3311†‡§¶	+14.63%	0.1555†‡§¶	+19.77%	0.4671†‡§¶	+16.50%	32,592/50,428/16,980
KESM	**0.5169**†‡§¶	+17.96%	**0.3336**†‡§¶	+15.47%	**0.1585**†‡§¶	+22.09%	**0.4713**†‡§¶	+17.53%	32,420/52,090/15,490

6 SALIENCE EVALUATION RESULTS

This section first presents the overall evaluation results for the entity salience task. Then it analyzes the advantages of modeling salience over counting frequency.

6.1 Entity Salience Performance

Table 3 shows the experimental results for the entity salience task. Frequency provides reasonable estimates of entity salience. The most frequent entity is often salient to the document; the Precision@1 is rather high, especially on the New York Times corpus. PageRank barely improves Frequency, although its embeddings are trained by the salience labels. LeToR, on the other hand, significantly improves both Precision and Recall of Frequency [9], which is expected as it has much richer features from various sources.

KESM outperforms all baselines significantly. Its improvements over LeToR are more than 10% on both datasets with only one exception: Precision@1 on New York Times. The improvements are also robust: About twice as many documents are improved (Win) than hurt (Loss).

We also conducted ablation studies on the source of evidence in KESM. Those marked with (E) include the entity kernels; those with (W) include word kernels; those with (K) enrich the entity embeddings with description embeddings. All variants include the entity kernels (E); otherwise the performances significantly dropped in our experiments.

KESM performs better than all of its variants, showing that all three sources contributed. Individually, KESM (E) outperforms all baselines. Compared to PageRank, the only difference is that KESM (E) uses kernels to model the interactions which are much more powerful than the raw embedding similarities used in PageRank [27]. KESM (EW) always significantly outperforms KESM (E). The interaction between an entity and document words conveys useful information, the distributed representations make them easily comparable, and the kernels model the word-entity interactions effectively. Knowledge enrichment (K) provides mixed results. A possible reason is that the training data is large enough to train good entity embeddings. Nevertheless, we find that adding the external knowledge makes the model stable and converged faster.

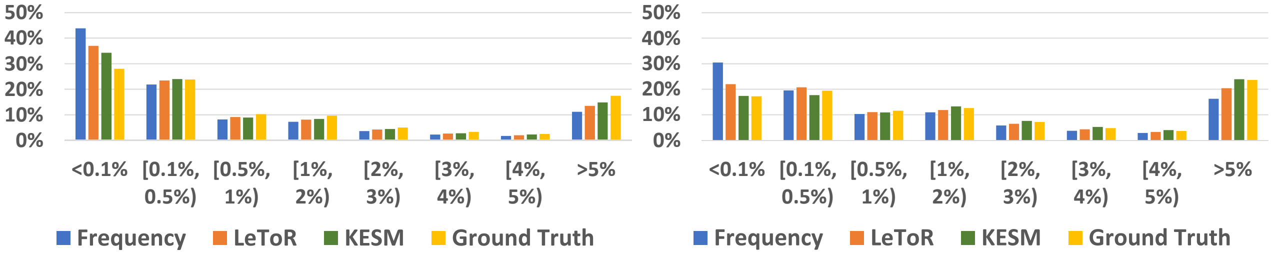
6.2 Modeling Salience VS. Counting Frequency

This experiment provides two analyses that study the advantage of KESM over counting frequency.

Ability to Model Tail Entities. The first advantage of KESM is that it is able to model the salience of less frequent (tail) entities. To demonstrate this effect, Figure 3 illustrates the distribution of predicted-salient entities in different frequency ranges. The entities with top k highest predicted scores are predicted-salient, while k is the number of salient entities in the ground truth.

In both datasets, the frequency-based methods are highly biased towards the head entities: The top 0.1% most popular entities receive almost two-times more salience predictions from Frequency than in ground truth. This is an intrinsic bias of frequency-based methods which not only limits their effectiveness but also attractiveness—less unexpected entities are selected.

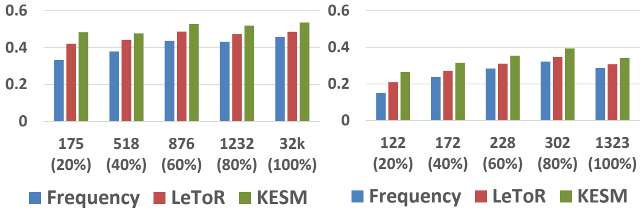
In comparison, the distributions of KESM are much closer to the ground truth. KESM does a better job in modeling tail entities because it estimates salience not only by frequency but also by modeling the *interactions* between entities and words. A tail entity can be estimated salient if many other entities and words in the



(a) New York Times

(b) Semantic Scholar

Figure 3: The distribution of salient entities predicted by different models. The entities are binned by their frequencies in testing data. The bins are ordered from most frequent (Top 0.1%) to less frequent (right). The x-axes mark the percentile range of each group. The y-axes are the fraction of salient entities in each bin. The histograms are ordered the same as the legends.



(a) New York Times

(b) Semantic Scholar

Figure 4: Performances on documents with varying lengths (number of words). The x-axes are the maximum length of the documents and the percentile of each group. The y-axes mark the performances on Precision@5. The histograms are ordered the same as the legends.

document are closely related to it. For example, there are many entities and words describing various aspects of an entity in its Wikipedia page; the entities and words on a personal homepage are probably related to the person. These entities and words can 'vote up' the title entity or the person because they are strongly connected to it/her. The ability to model such interactions with distributed representations and kernels is the main source of KESM's text understanding capability.

Reliable on Short Documents. The second advantage of KESM is its reliability on short texts. To demonstrate it, we analyzed the performances of models on documents of varying lengths. Figure 4 groups the testing documents into five bins by their lengths (number of words), ordered from short (left) to long (right). Their upper bounds and percentiles are marked on the x-axes. The Precision@5 of corresponding methods are marked on the y-axes.

Both Frequency and LeToR (whose features are also mostly frequency-based) are less reliable on shorter documents. The advantages of KESM are more significant when documents are shorter, while even in the longest bins where documents have thousands of words, KESM still outperforms Frequency and LeToR. Solely counting frequency is not sufficient to understand documents. The interactions between words and entities provide richer evidence and help KESM perform more reliably on shorter documents.

7 EXPERIMENTAL METHODOLOGY FOR AD HOC SEARCH

This section presents the experimental methodology for the ad hoc search task. It follows a popular setup in recent entity-oriented search research [26][2].

Datasets are from the TREC Web Track ad hoc search tasks, a widely used search benchmark. It includes 200 queries for the ClueWeb09 corpus and 100 queries for the ClueWeb12 corpus. The 'Category B' subsets of the two corpora and corresponding relevance judgments are used.

The ClueWeb09-B rankings re-ranked the top 100 documents retrieved by sequential dependency model (SDM) queries [17] with standard post-retrieval spam filtering [7]. On ClueWeb12-B13, SDM queries are not better than unstructured queries, and spam filtering provides mixed results; thus, we used unstructured queries and no spam filtering on this dataset, as in prior research [26]. All documents were parsed by Boilerpipe to title and body fields [13]. The query and document entities are from Freebase and were annotated by TagMe [11]. All entities are kept. It leads to high coverage and medium precision, the best setting found in prior research [25].

Evaluation Metrics are NDCG@20 and ERR@20, official evaluation metrics of TREC Web Tracks. Statistical significances are tested by permutation test (randomization test) with $p < 0.05$.

Baselines: The goal of our experiments is to explore the usage of entity salience modeling in ad hoc search. To this purpose, our experiments focus on evaluating the effectiveness of KESM's entity salience features in standard learning to rank; the proper baselines are the ranking features from word-based matches (IRFusion) and entity-based matches (ESR [29]). Unsupervised retrieval with words (BOW) and entities (BOE) are also included.

BOW is the base retrieval model, which is SDM on ClueWeb09-B and Indri language model on ClueWeb12-B.

BOE is the frequency-based retrieval with bag-of-entities [26]. It uses TagMe annotations and exact-matches query and documents in the entity space. It performs similarly to the entity language model [20] as they use the same information.

IRFusion uses standard word-based IR features such as language model, BM25, and TFIDF, applied to body and title fields. It is obtained from previous research [26].

[2] Available at http://boston.lti.cs.cmu.edu/appendices/SIGIR2017_word_entity_duet/

Table 4: Ad hoc search accuracy of KESM when used as ranking features in learning to rank. Relative performances over IRFusion are shown in the percentages. W/T/L are the number of queries a method improves, does not change, or hurts, compared with IRFusion. †, ‡, §, and ¶ mark the statistically significant improvements over BOE†, IRFusion‡, ESR§, and ESR+IRFusion¶. BOW is the base retrieval model, which is SDM in ClueWeb09-B and language model in ClueWeb12-B13.

	ClueWeb09-B					ClueWeb12-B13				
Method	NDCG@20		ERR@20		W/T/L	NDCG@20		ERR@20		W/T/L
BOW	0.2496	−5.26%	0.1387	−10.20%	62/38/100	0.1060	−12.02%	0.0863	−6.67%	35/22/43
BOE	0.2294	−12.94%	0.1488	−3.63%	74/25/101	0.1173	−2.64%	0.0950	+2.83%	44/19/37
IRFusion	0.2635	–	0.1544	–	–/–/–	0.1205	–	0.0924	–	–/–/–
ESR	0.2695†	+2.30%	0.1607	+4.06%	80/39/81	0.1166	−3.22%	0.0898	−2.81%	30/23/47
KESM	0.2799†	+6.24%	0.1663	+7.68%	85/35/80	0.1301†§	+7.92%	**0.1103**‡§¶	+19.35%	43/25/32
ESR+IRFusion	0.2791†‡	+5.92%	0.1613	+4.46%	91/34/75	0.1281	+6.30%	0.0951	+2.87%	45/24/31
KESM+IRFusion	**0.2993**†‡§¶	+13.58%	**0.1797**†‡§¶	+16.38%	98/35/67	**0.1308**†§	+8.52%	0.1079‡§¶	+16.77%	43/23/34

Table 5: Ranking performances of IRFusion, ESR, and KESM with title or body field individually. Relative performances (percentages) and Win/Tie/Loss are calculated by comparing with IRFusion on the same field. † and ‡ mark the statistically significant improvements over IRFusion† and ESR‡, also on the same field.

	ClueWeb09-B					ClueWeb12-B13				
Method	NDCG@20		ERR@20		W/T/L	NDCG@20		ERR@20		W/T/L
IRFusion-Title	0.2584	−3.51%	0.1460	−5.16%	83/48/69	0.1187	+6.23%	0.0894	+3.14%	41/23/36
ESR-Title	0.2678	–	0.1540	–	–/–/–	0.1117	–	0.0867	–	–/–/–
KESM-Title	**0.2780**†	+3.81%	**0.1719**†‡	+11.64%	91/46/63	**0.1199**	+7.36%	**0.0923**	+6.42%	35/28/37
IRFusion-Body	0.2550	+0.48%	0.1427	−3.44%	80/46/74	0.1115	+4.61%	0.0892	−3.51%	36/30/34
ESR-Body	0.2538	–	0.1478	–	–/–/–	0.1066	–	0.0924	–	–/–/–
KESM-Body	**0.2795**†‡	+10.13%	**0.1661**†‡	+12.37%	96/39/65	**0.1207**‡	+13.25%	**0.1057**†‡	+14.44%	43/24/33

ESR is the entity-based ranking features obtained from previous research [26]. It includes both exact and soft match signals in the entity space [29]. The differences with KESM are that in ESR, the query and documents are represented by frequency-based bag-of-entities [29] and the entity embeddings are pre-trained in the relation inference task [4].

Implementation Details: As discussed in Section 4, the TREC benchmarks do not have sufficient relevance labels for effective end-to-end learning; we pre-trained the KEE and KIM of KESM using the New York Time corpus and used them to extract salience ranking features. The entity salience features are combined by the same learning to rank model (RankSVM [12]) as used by IRFusion and ESR, with the same cross validation setup [26]. Similar to ESR, the base retrieval score is included as a feature in KESM. In addition, we also concatenate the features of ESR or KESM to IRFusion to evaluate their effectiveness when combined with word-based features. The resulting feature sets ESR+IRFusion and KESM+IRFusion were evaluated exactly the same as they were individually.

As a result, the comparisons of KESM with LeToR and ESR hold out all other factors and directly investigate the effectiveness of the salience ranking features in a widely used learning to rank model (RankSVM). Given the current exploration stage of entity salience in information retrieval, we believe this is more informative than mixing entity salience signals into more sophisticated ranking systems [23, 26], in which many other factors come into play.

8 SEARCH EVALUATION RESULTS

This section presents the evaluation results and case study in the ad hoc search task.

8.1 Overall Result

Table 4 lists the ranking evaluation results. The three supervised methods, IRFusion, ESR, and KESM, all use the exact same learning to rank model (RankSVM) and only differ in their features. ESR+IRFusion and KESM+IRFusion concatenate the two feature groups and use RankSVM to combine them.

On both ClueWeb09-B and ClueWeb12-B13, KESM features are more effective than IRFusion and ESR features. On ClueWeb12-B13, KESM individually outperforms other features significantly by 8 – 20%. On ClueWeb09-B, KESM provides more novel ranking signals; KESM+IRFusion significantly outperforms ESR+IRFusion. The fusion on ClueWeb12-B13 (KESM+LeToR) is not as successful perhaps because of the limited ranking labels on ClueWeb12-B13.

To better investigate the effectiveness of entity salience in search, we evaluated the features on individual document fields. Table 5 shows the ranking accuracies of the three feature groups when only the title field (Title) or the body field (Body) is used. As expected, KESM is more effective on the body field than on the title field: Titles are less noisy and perhaps all title entities are salient—not much new information is provided by salience modeling; on the other hand, body texts are longer and more complicated, providing more opportunities for better text understanding.

Table 6: Examples from queries that KESM improved or hurt, compared to ESR. Documents are selected from those that ESR and KESM disagreed. The descriptions are manually written to reflect the main topics of the documents.

Cases that KESM Improved			
Query	**Query Entities**	**ESR Preferred Document**	**KESM Preferred Document**
ER TV Show	“ER (TV Series)” “TV Program”	clueweb09-enwp02-22-20096 “List of films in Wiki without article”	clueweb09-enwp00-55-07707 “ER (TV series) - Wikipedia”
Wind Power	“Wind Power ”	clueweb12-0200wb-66-32730 “Home solar power systems”	clueweb12-0009wb-54-01932 “Wind energy — Alternative Energy HQ”
Hurricane Irene Flooding in Manville NJ	“Hurricane Irene” “Flood”; “Manville, NJ”	clueweb12-0705wb-49-04059 “Disaster funding for Hurricane Irene”	clueweb12-0715wb-81-29281 “Videos and news about Hurricane Irene”

Cases that KESM Hurt			
Query	**Query Entities**	**ESR Preferred Document**	**KESM Preferred Document**
Fickle Creek Farm	“Malindi Fickle” “Stream”; “Farm”	clueweb09-en0003-97-27345 “Hotels near Fickle Creak”	clueweb09-en0005-66-00576 “List of breading farms”
Illinois State Tax	“Illinois”; “State Government” “US Tax”	clueweb09-enwp01-67-20725 “Sales taxes in the United States, Wikipedia”	clueweb09-en0011-23-05274 “Retirement-related general purpose taxes by State”
Battles in the Civil War	“Battles” “Civil War”	clueweb09-enwp03-20-07742 “List of American Civil War battles”	clueweb09-enwp01-30-04139 “List of wars in the Muslim world”

The salience ranking features also behave differently with ESR and IRFusion. As shown by the W/T/L ratios in Table 4 and Table 5, more than 70% query rankings are changed by KESM. The ranking evidence provided by KESM features is from the interactions of query entities with the entities and words in the candidate documents. This evidence is learned from the entity salience corpus and is hard to be described by traditional frequency-based features.

8.2 Case Study

The last experiment provides case studies on how KESM transfers its text understanding ability to search, by comparing the rankings of KESM-Body with ESR-Body. Both ESR and KESM match query and documents in the entity space, but ESR uses frequency-based bag-of-entities to represent documents while KESM uses entity salience. We picked the queries where KESM-Body improved or hurt compared to ESR-Body and manually examined the documents they disagreed. The examples are listed in Table 6.

The improvements from KESM are mainly from its ability to determine whether a candidate document emphasizes the query entities or just mentions the query terms. As shown in the top half of Table 6, KESM promotes documents where the query entities are more salient: the Wikipedia page about the ER TV show, a homepage about wind power, and a news article about the hurricane. On the other hand, ESR's frequency-based ranking might be confused by web pages that only partially talk about the query topic. It is hard for ESR to exclude those web pages because they also mention the query entities multiple times.

Many errors KESM made are due to the lack of text understanding on the query side. KESM focuses on modeling the salience of entities in the *candidate documents* and its ranking model treats all query entities equally. As shown in the lower half of Table 6, the query entities may contain errors, for example, “Malindi Fickle”, or general entities that blur the (perhaps implied) query intent, for example “Civil War”, “State government”, and “US Tax’. These query entities do not align well with the information needs and thus mislead KESM. Modeling the entity salience in *queries* is a different task which is more about understanding search intents. To address these error cases may require a deeper fusion of KESM in more sophisticated ranking systems that can handle noisy query entities [26, 28].

9 CONCLUSION

This paper presents KESM, the Kernel Entity Salience Model that estimates the salience of entities in documents. KESM represents entities and words with distributed representations, models their interactions using kernels, and combines the kernel scores to estimate entity salience. The semantics of entities in the knowledge graph—their descriptions—are also incorporated to enrich entity embeddings. In the entity salience task, the whole model is trained end-to-end using automatically generated salience labels.

In addition to the entity salience task, KESM is also applied to ad hoc search and ranks documents by the salience of query entities in them. It calculates the kernel scores of query entities in the document, combines them to salience ranking features, and uses a ranking model to predict the query-document ranking score. When ranking labels are scarce, the ranking features can be extracted by pre-trained distributed representations and kernels from the entity salience task and then used by standard learning to rank. These ranking features convey KESM's text understanding ability learned from entity salience labels to search.

Our experiments on two entity salience corpora, a news corpus (New York Times) and a scientific publication corpus (Semantic Scholar), demonstrate the effectiveness of KESM in the entity salience task. Significant and robust improvements are observed over frequency and feature-based methods. Compared to those baselines, KESM is more robust on tail entities and shorter documents; its Kernel Interaction Model is more powerful than the raw embedding similarities in modeling term interactions. Overall, KESM is a stronger model with a more powerful architecture.

Our experiments on ad hoc search were conducted on the TREC Web Track queries and two ClueWeb corpora. In both corpora, the salience features provided by KESM trained on the New York Times corpus outperform both word-based ranking features and frequency-based entity-oriented ranking features, despite differences between the salience task and the ranking task. The advantages of the salience features are more observed on the document bodies on which deeper text understanding is required.

Our case studies on the winning and losing queries of KESM illustrate the influences of the salience ranking features: they distinguish documents in which the query entities are the core topic from those where the query entities are only partial to their central ideas. Interestingly, this leads to both winning cases—better text understanding leads to more accurate search—and also losing cases: when the query entities do not align well with the underlying search intent, emphasizing them ends up misleading the document ranking.

We find it very encouraging that KESM successfully transfers the text understanding ability from entity salience estimation to search. Estimating entity salience is a fine-grained text understanding task that focuses on the detailed interactions between entities and words. Previously it was uncommon for text processing techniques at this granularity to be as effective in information retrieval. Often shallower methods worked better for search. However, the fine-grained text understanding provided by KESM—the interaction and consistency between query entities with the document entities and words—actually improves the ranking accuracy. We view this work as an encouraging step from "search by matching" to "search with meanings" [1] and hope it will motivate more future explorations towards this direction.

10 ACKNOWLEDGMENTS

This research was supported by National Science Foundation (NSF) grant IIS-1422676 and DARPA grant FA8750-12-2-0342 under the DEFT program. Any opinions, findings, and conclusions in this paper are the authors' and do not necessarily reflect the sponsors'.

REFERENCES

[1] Hannah Bast, Björn Buchhold, Elmar Haussmann, and others. 2016. Semantic search on text and knowledge bases. *Foundations and Trends in Information Retrieval* 10, 2-3 (2016), 119–271.

[2] Michael Bendersky, Donald Metzler, and W. Bruce Croft. 2011. Parameterized concept weighting in verbose queries. In *Proceedings of the 34th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2011)*. ACM, 605–614.

[3] Roi Blanco and Christina Lioma. 2012. Graph-based term weighting for information retrieval. *Information Retrieval* 15, 1 (2012), 54–92.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NIPS 2013)*. NIPS, 2787–2795.

[5] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search Engines: Information Retrieval in Practice.* Addison-Wesley Reading.

[6] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018)*. ACM, 126–134.

[7] Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2014)*. ACM, 365–374.

[8] Milan Dojchinovski, Dinesh Reddy, Tomás Kliegr, Tomas Vitvar, and Harald Sack. 2016. Crowdsourced Corpus with Entity Salience Annotations.. In *Proceedings of the 10th Edition of the Languge Resources and Evaluation Conference (LREC 2016)*.

[9] Jesse Dunietz and Daniel Gillick. 2014. A New Entity Salience Task with Millions of Training Examples.. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*. ACL, 205–209.

[10] Faezeh Ensan and Ebrahim Bagheri. 2017. Document retrieval model through semantic linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM 2017)*. ACM, 181–190.

[11] Paolo Ferragina and Ugo Scaiella. 2010. Fast and accurate annotation of short texts with Wikipedia pages. *arXiv preprint arXiv:1006.3498* (2010).

[12] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002)*. ACM, 133–142.

[13] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web Search and Data Mining (WSDM 2010)*. ACM, 441–450.

[14] Tie-Yan Liu. 2009. Learning to rank for Information Retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.

[15] Xitong Liu and Hui Fang. 2015. Latent entity space: A novel retrieval approach for entity-bearing queries. *Information Retrieval Journal* 18, 6 (2015), 473–503.

[16] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*. ACL, 55–60.

[17] Donald Metzler and W Bruce Croft. 2005. A Markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*. ACM, 472–479.

[18] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing (EMNLP 2004)*. ACL, 404–411.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Advances in Neural Information Processing Systems 2013 (NIPS 2013)*. NIPS, 3111–3119.

[20] Hadas Raviv, Oren Kurland, and David Carmel. 2016. Document retrieval using entity-based language models. In *Proceedings of the 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*. ACM, 65–74.

[21] François Rousseau and Michalis Vazirgiannis. 2013. Graph-of-word and TW-IDF: New approach to ad hoc IR. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management (CIKM 2013)*. ACM, 59–68.

[22] Evan Sandhaus. 2008. The New York Times annotated corpus. *Linguistic Data Consortium, Philadelphia* 6, 12 (2008), e26752.

[23] Chenyan Xiong and Jamie Callan. 2015. EsdRank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM 2015)*. ACM, 951–960.

[24] Chenyan Xiong and Jamie Callan. 2015. Query expansion with Freebase. In *Proceedings of the fifth ACM International Conference on the Theory of Information Retrieval (ICTIR 2015)*. ACM, 111–120.

[25] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2016. Bag-of-Entities representation for ranking. In *Proceedings of the sixth ACM International Conference on the Theory of Information Retrieval (ICTIR 2016)*. ACM, 181–184.

[26] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-entity duet representations for document ranking. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, 763–772.

[27] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, 55–64.

[28] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Eduard H. Hovy. 2017. JointSem: Combining query entity linking and entity based document ranking. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017)*. 2391–2394.

[29] Chenyan Xiong, Russell Power, and Jamie Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. ACM, 1271–1279.

[30] Le Zhao and Jamie Callan. 2010. Term necessity prediction. In *Proceedings of the 19th ACM International on Conference on Information and Knowledge Management (CIKM 2010)*. ACM, 259–268.

[31] Guoqing Zheng and James P. Callan. 2015. Learning to reweight terms with distributed representations. In *Proceedings of the 38th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2015)*. ACM, 575–584.