

如何使用知识图谱增强信息检索模型？

原创 刘布楼 PaperWeekly 2020-02-26

©PaperWeekly 原创 · 作者 | 刘布楼

学校 | 清华大学博士生

研究方向 | 信息检索

传统的信息检索模型中，文本通常使用词袋模型表示。该方法有两个较为明显的缺陷：1) 只能通过 TF-IDF 等相关信号判断查询-文本相关性；2) 模型没有深入理解查询和文本的语义信息，而是更多地依赖于特征工程的方法。

近年来流行的方法主要可以分为两大类：1) 通过知识图谱引入背景信息和先验知识；2) 通过深度学习技术从大规模的数据中学到信息的隐性表示。

本文主要针对第一种思路进行探讨：通过引入知识图谱中的实体以及实体的描述信息丰富语义，从而优化信息检索模型。知识图谱中的实体可以是一些概念、人名地名等等，这些实体排除介词的干扰，赋予查询和文本更简洁的表示，并引入知识图谱中的语义信息，增强信息检索模型的语义理解能力。

本文内容将介绍**五篇通过知识图谱优化信息检索模型的论文**，分别是：1) 基于知识图谱的学术检索模型；2) 词袋和实体的结合表示优化排序；3) 结合查询实体链接特征优化排序；4) 基于核方法的实体重要性建模和排序优化；5) 基于实体的神经信息检索模型。

01

基于知识图谱的学术检索模型

Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding

Chenyan Xiong*
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
cx@cs.cmu.edu

Russell Power
Allen Institute for
Artificial Intelligence
Seattle, WA, USA
russellp@allenai.org

Jamie Callan
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
callan@cs.cmu.edu

论文链接: http://www.cs.cmu.edu/~cx/papers/Explicit_Semantic_Ranking.pdf

论文概述

当前学术检索系统面临的主要挑战是检索系统无法理解学术概念而限制了学术检索的效果。例如“Softmax Categorization”和“Softmax Classification”表达的含义相同，但是词袋模型无法将其归为一类；而“Dynamic programming segmentation”在图像处理领域中表示语义分割，在自然语言处理领域中表示分词，但是词袋模型无法区分这两种概念。

本提出的方法是借助知识图谱，在实体空间中对查询和文本进行表示，然后通过他们的知识图谱嵌入表示建立语义连接，从而优化查询的效果。

算法详解

该算法主要分为两个部分，第一部分是建立知识图谱，第二部分是根据查询对学术论文进行语义排序。

建立知识图谱 (Knowledge Graph Construction) :

建立知识图谱首先需要获取实体，论文给出两种途径：从 S2's corpus 中抽取关键词和从 freebase 里面获得；然后将论文中出现的 surface form 链接到实体；再后对每个实体建立四种边，分别是作者 (author), 上下文 (context), 描述 (desc), 发表刊物 (venue); 最后根据这四种边所连接的对象分别用 skip-gram 的方法得到实体嵌入表示。

语义排序 (Explicit Semantic Ranking) :

该部分首先将查询和文本链接的实体两两计算语义相似度：

$$s(e_i, e_j) = \cos(V(e_i), V(e_j)).$$

然后在查询词维度上做 max-pooling 操作（相当于针对每个文本实体仅考虑和它语义相关性最大的查询实体）：

$$\vec{S}(d) = \max_{e_i \in \vec{E}_q} s(e_i, \vec{E}_d).$$

再后在文本维度上做 bin-pooling 操作（将不同范围内的语义匹配信号各自相加，从而概括查询实体和文本实体之间的匹配情况）：

$$B_k(q, d) = \log \sum_j I(st_k \leq \vec{S}_j(d) < ed_k).$$

最后以此为特征输入全连接层计算得到最后的分数：

$$f(q, d|G) = w_0 f_{S2}(q, d) + W^T B(q, d)$$

具体的算法流程如图所示：

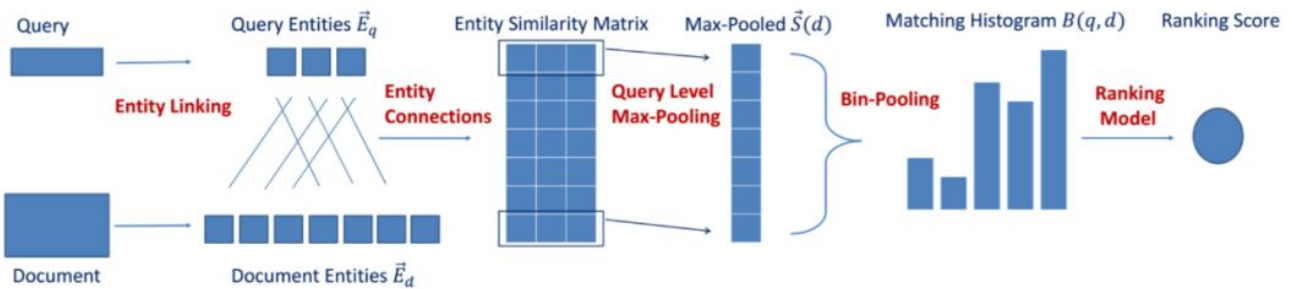


Figure 2: The Framework of Explicit Semantic Ranking (ESR).

创新和发现

建立辅助学术检索的知识图谱，包括两个重要发现：使用的 edge2vec 的方法是一种效率高且效果好的知识图谱嵌入表示方法；使用 freebase 获得的实体相比于直接从 S2's corpus 中抽取关键词效果更好。

使用 bin-pooling 操作抽取聚合相关性信号：该方法结合了直接匹配 (exact matching) 信号和软匹配 (soft matching) 信号，并将不同范围内的语义匹配信号各自相加，更有效的发掘查询和文本之间的语义相关信息。



词袋和实体的结合表示优化排序

Word-Entity Duet Representations for Document Ranking

Chenyan Xiong
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
cx@cs.cmu.edu

Jamie Callan
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
callan@cs.cmu.edu

Tie-Yan Liu
Microsoft Research
Beijing 100080, P.R. China
tie-yan.liu@microsoft.com

论文链接: <https://arxiv.org/abs/1706.06636>

论文概述

在 ESR 论文中，查询和文本的表示仅考虑了链接实体而忽略了原有的词，导致了一定的信息损失。

本文提出了一种将词袋和知识图谱链接到的实体相结合，从而优化信息检索的框架。采用的方法是首先通过基于词袋的表示和基于实体的表示对查询和文本建模，然后利用词袋空间交互信息、实体空间交互信息和跨空间交互信息生成排序特征，最后使用注意力机制消除实体注释过程中产生的噪音信息，生成最终的相关性分数。

算法详解

该算法的主要流程分为三个部分，即第一部分是生成基于词袋和实体的表示，第二部分是词袋-实体表示结合的相关性抽取，第三部分是注意力机制聚合相关性信息。

生成基于词袋和实体的表示 (word and entity based representations):

该部分将查询和文本表示为词袋和实体两种形式（即每次词/实体及其出现的频率）。其中实体的筛选过程包括两个步骤：获取 surface form (spotting) 和选取最有可能的候选实体 (disambiguation)。

词袋-实体表示结合的相关性抽取 (matching with word-entity duet):

该部分首先基于四种交互方式（查询词-文本词、查询实体-文本词、查询词-文本实体、查询实体-文本实体）生成特征，前三者主要基于统计特征，实体-实体交互特征基于 ESR 模型生成的排序特征。

查询词-文本词具体构造方法：

Table 1: Ranking features from query words to document words (title and body) (Φ_{QW-DW}).

Feature Description	Dimension
BM25	2
TF-IDF	2
Boolean OR	2
Boolean And	2
Coordinate Match	2
Language Model (Lm)	2
Lm with JM smoothing	2
Lm with Dirichlet smoothing	2
Lm with two-way smoothing	2
Total	18

查询实体-文本词具体构造方法：

Table 2: Ranking features from query entities (name and description) to document words (title and body) (Φ_{Qe-Dw}).

Feature Description	Dimension
BM25	4
TF-IDF	4
Boolean Or	4
Boolean And	4
Coordinate Match	4
Lm with Dirichlet Smoothing	4
Total	24

查询词-文本实体具体构造方法:

Table 3: Ranking features from query words to document entities (name and description) (Φ_{Qw-De}).

Feature Description	Dimension
Top 3 Coordinate Match on Title Entities	6
Top 5 Coordinate Match on Body Entities	10
Top 3 TF-IDF on Title Entities	6
Top 5 TF-IDF on Body Entities	10
Top 3 Lm-Dirichlet on Title Entities	6
Top 5 Lm-Dirichlet on Body Entities	10
Total	48

查询实体-文本实体具体构造方法:

Table 4: Ranking features from query entities to document’s title and body entities (Φ_{Qe-De}).

Feature Description	Dimension
Binned translation scores, 1 exact match bin, 5 soft match Bins in the range [0, 1).	12

注意力机制聚合相关性信息 (attention features):

针对于查询实体可能不准确的问题，该方法针对交互特征使用注意力机制生成注意力特征（即各个特征的权重），该注意力特征主要从歧义特征和查询重要性两个方面生成，具体表示如下：

Table 5: Attention features for query entities.

Feature Description	Dimension
Entropy of the Surface Form	1
Is the Most Popular Candidate Entity	1
Margin to the Next Candidate Entity	1
Embedding Similarity with Query	1
Total	4

因此对查询词交互特征和查询实体相关特征进行区分，查询词注意力分数设置为 1，查询实体注意力分数根据注意力特征获得，具体表示如下：

$$R_w(w_i, \cdot) = \Phi_{Q_w-D_w}(w_i) \sqcup \Phi_{Q_w-D_e}(w_i)$$

$$R_e(e_j, \cdot) = \Phi_{Q_e-D_w}(e_j) \sqcup \Phi_{Q_e-D_e}(e_j)$$

$$A_w(w_i, \cdot) = 1$$

$$A_e(e_j, \cdot) = \text{Att}(e_j).$$

然后使用一维 CNN 聚合交互特征和注意力特征，并将二者相乘得到最后的相关性分数：

$$F_w(w_i) = W_w^m \cdot R_w(w_i, \cdot) + b_w^m$$

$$F_e(e_j) = W_e^m \cdot R_e(e_j, \cdot) + b_e^m.$$

$$\alpha_w(w_i) = \text{ReLU}(W_w^a \cdot A_w(w_i, :) + b_w^a)$$

$$\alpha_e(e_j) = \text{ReLU}(W_e^a \cdot A_e(e_j, :) + b_e^a).$$

$$f(q, d) = F_w \cdot \alpha_w + F_e \cdot \alpha_e.$$

具体算法流程图如图所示：

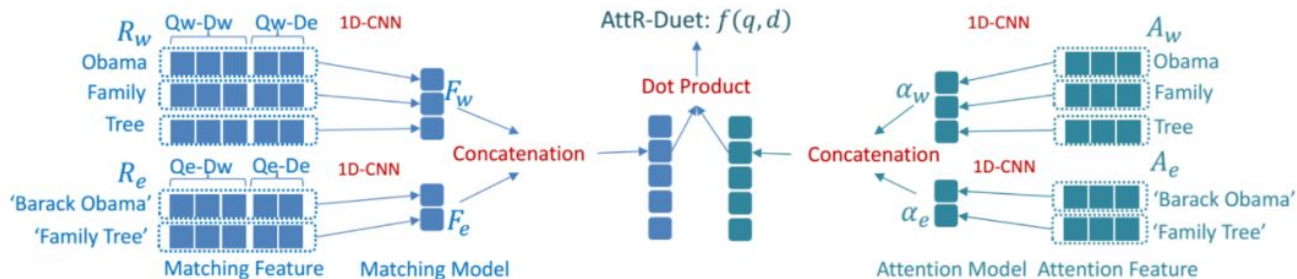


Figure 1: The Architecture of the Attention based Ranking Model for Word-Entity Duet (AttR-Duet).

创新和发现

将词袋表示和实体表示相结合，克服了仅适用词袋表示时语义信息不丰富和仅适用实体表示时造成信息损失的缺点，优化了知识图谱增强信息检索模型的结构。

针对查询实体引入了注意力特征，为生成的排序信号赋予权重，从而解决了引入的查询实体不准确的问题，特别在查询实体较多的情况下体现出较好的效果。



结合查询实体链接特征优化排序

JointSem: Combining Query Entity Linking and Entity based Document Ranking

Chenyan Xiong, Zhengzhong Liu, Jamie Callan, Eduard Hovy
 Language Technologies Institute
 Carnegie Mellon University
 Pittsburgh, PA 15213
 cx,liu,callan,hovy@cs.cmu.edu

论文链接: <http://www.cs.cmu.edu/~cx/papers/JointSem.pdf>

论文概述

在原有的方法构建排序特征的过程中，实体链接往往只是一个预处理的步骤，而没有 将链接的信息作为特征加入到排序模型里。

本文提出了一种将查询实体链接和基于实体的文本排序模型结合起来构造语义相似度排序模型的办法。采用的方法是利用获取 surface form (spotting) 的信息和链接 (linking) 信息表示候选实体的重要性，从而通过链接实体的过程优化文本排序的特征。

算法详解

该算法的流程相对较为简单，主要是分为两个步骤，第一步生成三种特征（surface form 特征，链接特征和基于实体的文本排序特征），第二步根据这些特征计算排序分数。特征生成（surface form 特征，链接特征和基于实体的文本排序特征）：

Surface form 特征： 首先将查询划分为能在 surface form 字典里面查找到的最大的 ngrams，然后根据每个 surface form 和候选实体之间的关系构造特征。例如和语料库中的每个实体分别连接的概率、连接概率最大的实体和第二大的实体之间概率的差值、自身的长度等等。

链接特征： 将每个实体对齐到第一步所得到的 surface form 的过程中获取。例如和每个 surface form 之间连接的概率，和查询词之间通过嵌入表示计算的相似度以及和每个查询词的最匹配实体之间通过嵌入表示计算的相似度等。

基于实体的文本排序特征： 通过普通的检索模型得到的文本排序特征，例如 BM25, TF-IDF 等等。

具体三种特征表示如下：

Table 1: Spotting, linking, and ranking features. Surface Form Features are extracted for each spotted surface form. Entity Features are extracted for each candidate entity from each spot. Entity-Document Ranking Features are extracted for each entity-document pair. The number in brackets is the dimension of the corresponding feature group.

Surface Form Features (ϕ_s)	Entity Features (ϕ_e)	Entity-Document Ranking Features (ϕ_r)
(1) Linked Probability	(1) Commonness	(16) BM25, Coordinate Match, TFIDF and language model with Dirichlet smoothing from entity's textual fields (name and description) to document's fields (title and body)
(1) Surface Form Entropy	(2) Max and Mean Similarity with Query Words	
(1) Top Candidate Entities Margin		
(1) Surface Form Length and Coverage	(2) Max and Mean Similarity with Other Query Entities	
(1) Surface Form Coverage		

相关性分数计算：

Surface form 的重要性表示如下：

$$f_s(s_i) = w_s^T \phi_s(s_i).$$

对齐实体的重要性表示如下：

$$f_e(e_j^i) = w_e^T \phi_e(e_j^i).$$

文本排序特征表示如下：

$$f_r(e_j^i, d) = w_r^T \phi_r(e_j^i, d).$$

最终的相关性分数计算方法如下：

$$f(q, d|\theta, S, E) = \sum_{i=1}^M \sum_{j=1}^N f_s(s_i) \cdot f_e(e_j^i) \cdot f_r(e_j^i, d),$$

创新和发现

本文通过抽取实体链接过程中的特征，赋予链接实体不同的权重，从而优化基于实体的文本排序模型。

针对 surface form 选取链接性靠前的几个实体，相比于只选取一个实体，排序的效果可以得到提升。



基于核方法的实体重要性建模和排序优化

Towards Better Text Understanding and Retrieval through Kernel Entity Saliency Modeling

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

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

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

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

论文链接: <https://arxiv.org/abs/1805.01334>

论文概述

在原来的工作中，实体重要性建模主要停留在查询实体的层面上，而文本实体重要性主要通过出现频率确定。

本文提出的方法更好地估计文本中的实体重要性。具体来说，首先使用知识图谱中的描述信息丰富实体表示，再利用核方法对目标实体和文本中的词、实体进行交互式建模，最后将交互式特征聚合估计实体重要性。

在该方法的基础上可以优化信息检索模型的效果。具体来说，可以将查询实体和目标文本中的词和实体以核方法进行交互式建模得到特征作为排序特征，从而计算相关性分数。

算法详解

该模型获取交互式特征的过程可以分为两个步骤：加入知识的实体表示 (Knowledge Enriched Embedding) 和基于核方法的交互式建模 (Kernel Interaction Model)。随后将该特征输入全连接网络可以用于实体重要性估计任务和信息检索任务。

加入知识的实体表示 (Knowledge Enriched Embedding):

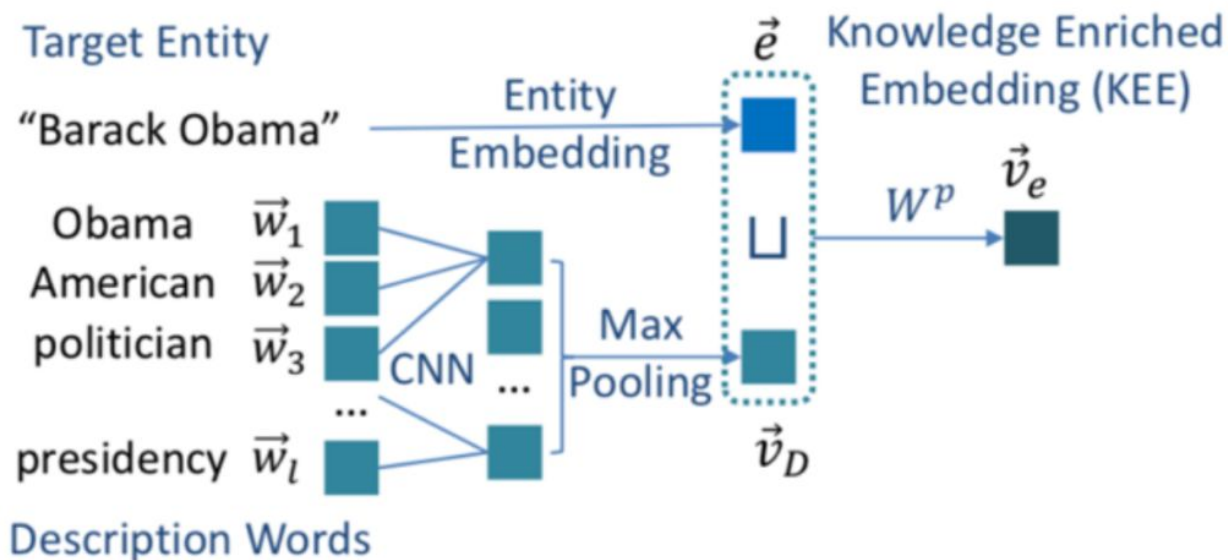
该步骤首先使用 CNN 对实体描述信息建模得到新的实体描述表示:

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

然后将实体表示和实体描述表示拼接并聚合得到加入知识的实体表示:

$$\vec{v}_e = W^p \cdot (\vec{e} \sqcup \vec{v}_D). \quad \text{KEE Embedding}$$

完整的加入知识的实体表示生成流程如下图所示:



(a) Knowledge Enriched Embedding (KEE)

基于核方法的交互式建模 (Kernel Interaction Model):

该步骤首先将目标实体和文本词使用核方法进行交互式建模：

$$\Phi(e_i, \mathbb{E}) = \{\phi_1(e_i, \mathbb{E}), \dots, \phi_k(e_i, \mathbb{E}), \dots, \phi_K(e_i, \mathbb{E})\},$$

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

然后将目标实体和文本实体使用核方法进行交互式建模：

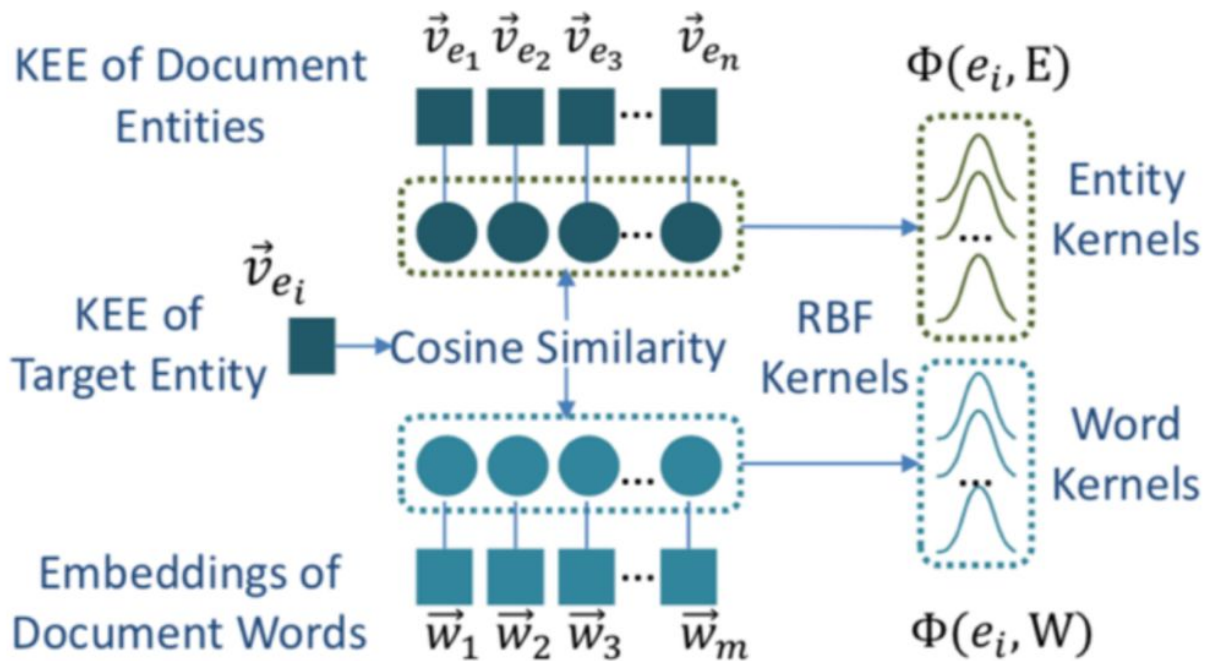
$$\Phi(e_i, \mathbb{W}) = \{\phi_1(e_i, \mathbb{W}), \dots, \phi_k(e_i, \mathbb{W}), \dots, \phi_K(e_i, \mathbb{W})\},$$

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

最后将二者拼接获得交互式特征：

$$KIM(e_i, d) = \Phi(e_i, \mathbb{E}) \sqcup \Phi(e_i, \mathbb{W}).$$

完整基于核方法的交互式建模流程如下：



(b) Kernel Interaction Model (KIM)

实体重要性打分：计算实体相对于文本的重要性程度只需将交互式特征输入全连接层即可：

$$f(e_i, d) = W^s \cdot KIM(e_i, d) + b^s.$$

检索模型应用：计算查询实体和文本的匹配程度只需将查询实体输入该模型建模得到交互式特征，并在交互式特征的每个维度上计算 $\log \text{sum}$ 得到统一的交互式特征，并输入全连接层得到最终的相关分数：

$$f(q, d) = W^r \cdot \Psi(q, d),$$

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

完整的检索模型应用流程图如下：

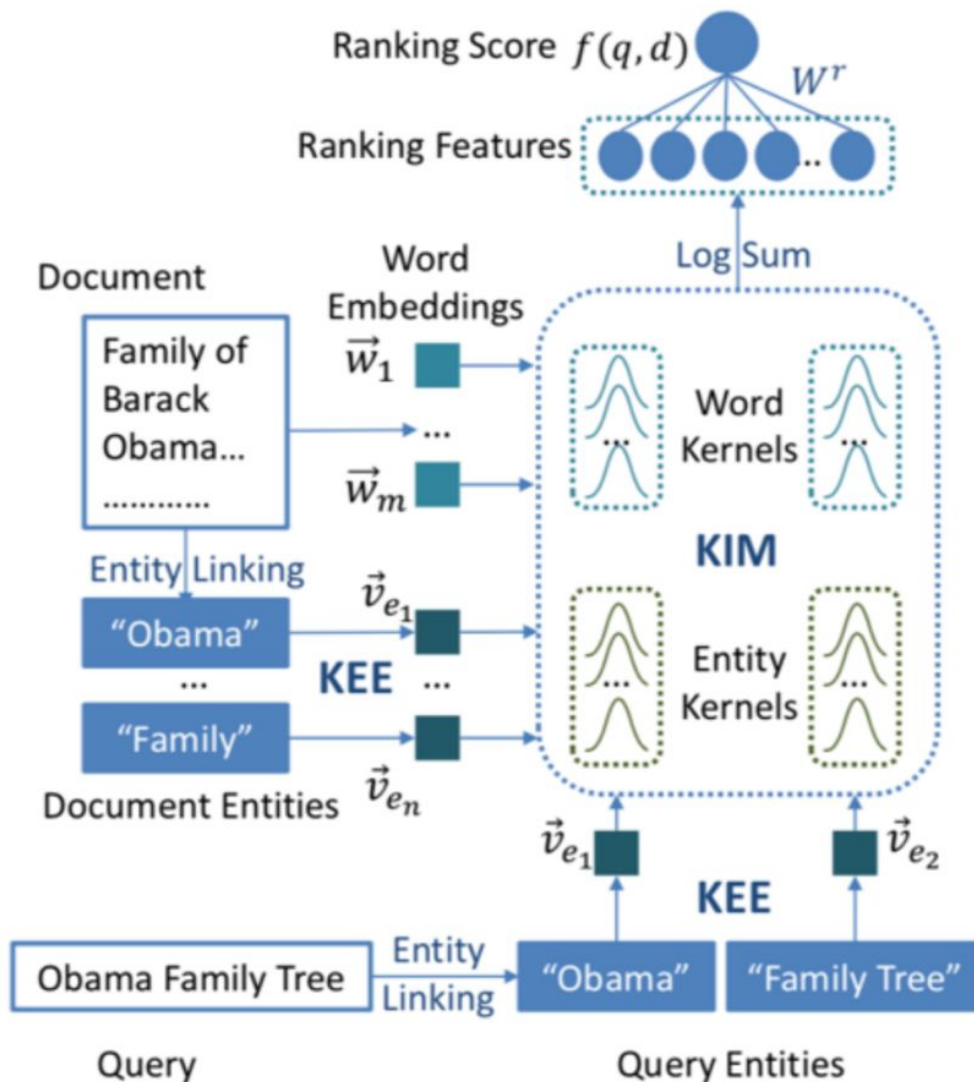


Figure 2: Ranking with KESM. KEE embeds the entities. KIM calculates the kernel scores of query entities VS. document entities and words. The kernel scores are combined to ranking features and then to the ranking score.

创新和发现

在频繁使用的实体和较少使用的实体之间具有较好的平衡型，这是对实体频率模型最重要的改进。

该方法在短文中的效果相比于原来方法有较大的提高。

在信息检索的过程中，可以辅助判断查询实体的重要性（提及还是强调），有利于增强模型对查询的理解。



基于实体的神经信息检索模型

Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval

Zhenghao Liu¹ Chenyan Xiong² Maosong Sun¹ * Zhiyuan Liu¹

¹State Key Laboratory of Intelligent Technology and Systems

Beijing National Research Center for Information Science and Technology

Department of Computer Science and Technology, Tsinghua University, Beijing, China

²Language Technologies Institute, Carnegie Mellon University

论文链接: <https://www.aclweb.org/anthology/P18-1223>

源码链接: <https://github.com/thunlp/EntityDuetNeuralRanking>

论文概述

在以上介绍的方法中，均是通过知识图谱提取人工特征，而没有利用神经网络自动提取知识图谱的语义信息。

本文将知识图谱中的实体信息作为先验知识加入神经信息检索已有的较好模型（KNRM [6], Conv-KNRM [7]）中，提出 EDRM 模型，从而达到提升信息检索泛化能力的效果。

算法详解

该方法分为两个部分，第一部分是生成四个相关信号矩阵：查询词-文本词矩阵、查询实体-文本词矩阵、查询词-文本实体矩阵、查询实体-文本实体矩阵，第二部分是结合 KNRM 和 Conv-KNRM 等神经信息检索模型聚合相关信号矩阵得到相关分数。

相关信号矩阵生成：

该步骤需要根据词向量和实体语义向量分别计算相似度矩阵。其中查询词嵌入和文本词嵌入的方法和神经信息检索的方法是完全一样的，如果使用 K-NRM 则直接使用原本的词向量，如果使用 Conv-KNRM 则根据卷积核的大小变化得到 unigrams, bigrams, trigrams 的向量。实体语义向量需要计算三个向量并聚合得到，分别是实体向量，描述向量和类型向量。

实体向量直接使用实体本身的向量即可：

$$\vec{v}_e^{\text{emb}} = \text{Emb}_e(e).$$

描述向量需要将实体描述的词向量通过卷积神经网络的计算得到：

$$\vec{g}_e^j = \text{ReLu}(W_{\text{CNN}} \cdot \vec{V}_w^{j:j+h} + \vec{b}_{\text{CNN}}),$$

$$\vec{v}_e^{\text{des}} = \max(\vec{g}_e^1, \dots, \vec{g}_e^j, \dots, \vec{g}_e^m).$$

类型向量需要将实体的类型向量通过注意力机制计算得到：

$$\vec{v}_{f_j}^{\text{emb}} = \text{Emb}_{\text{tp}}(e).$$

$$a_j = \frac{\exp(P_j)}{\sum_l^n \exp(P_l)},$$

$$P_j = \left(\sum_i W_{\text{bow}} \vec{v}_{t_i} \right) \cdot \vec{v}_{f_j}.$$

$$\vec{v}_e^{\text{type}} = \sum_j^n a_j \vec{v}_{f_j},$$

将三种向量聚合得到实体语义向量：

$$\vec{v}_e^{\text{sem}} = \vec{v}_e^{\text{emb}} + W_e(\vec{v}_e^{\text{des}} \oplus \vec{v}_e^{\text{type}})^T + \vec{b}_e.$$

最后计算四个相关信号矩阵：查询词-文本词矩阵、查询实体-文本词矩阵、查询词-文本实体矩阵、查询实体-文本实体矩阵，并拼接得到排序特征：

$$\begin{aligned} M_{ww}^{ij} &= \cos(\vec{v}_{wq}^i, \vec{v}_{wd}^j); M_{ee}^{ij} = \cos(\vec{v}_{eq}^i, \vec{v}_{ed}^j) \\ M_{ew}^{ij} &= \cos(\vec{v}_{eq}^i, \vec{v}_{wd}^j); M_{we}^{ij} = \cos(\vec{v}_{wq}^i, \vec{v}_{ed}^j). \end{aligned}$$

$$\Phi(\mathcal{M}) = \phi(M_{ww}) \oplus \phi(M_{we}) \oplus \phi(M_{ew}) \oplus \phi(M_{ee}),$$

神经信息检索模型聚合相关信号：

神经信息检索模型 K-NRM 通过核池化函数抽取不同范围内的相关信号，将每个相似度矩阵输入可以得到排序特征：

$$K_k(M) = \sum_j \exp\left(-\frac{M^{ij} - \mu_k}{2\delta_k^2}\right).$$

对于 Conv-KNRM 模型，需要构造基于 n-grams 的相似度矩阵：

$$\phi(M_{h_q, h_d}) = \phi(M_{ww^{h_q, h_d}}) \oplus \phi(M_{ew^1, h_d}) \oplus \phi(M_{we^{h_q, 1}}). \quad (17)$$

$$\Phi(\mathcal{M}) = \phi(M_{1,1}) \oplus \dots \oplus \phi(M_{h_q, h_d}) \oplus \dots \oplus \phi(M_{ee}).$$

最后将核池化函数提取的相关性特征作为全连接层的输入得到相关性分数：

$$f(q, d) = \tanh(\omega_r^T \Phi(\mathcal{M}) + b_r).$$

创新和发现

EDRM 模型具有较好的泛化能力，相较于原来的神经信息检索模型有更好的排序效果。

在信息较难提取的情况（如查询文本长度较短）下：EDRM 相较于原来的神经信息检索模型有较大突破，这说明该模型可以在信息较少的情况下结合知识图谱中的信息提高查询的效果。

总结

通过知识图谱引入背景信息和先验知识已经在信息检索模型中取得了较好的效果，证明实体语义对于理解查询意图、优化排序结果有很大的帮助，未来的研究可以从引入知识图谱的关系入手，建立更丰富、更智能化的知识指导式信息检索模型，而不仅仅局限于知识图谱中的实体语义信息。

参考文献

- [1] Chenyan Xiong, Russell Power, Jamie Callan. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. WWW (2017)
- [2] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Word-entity duet representations for document ranking. SIGIR (2017)
- [3] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, Eduard Hovy. JointSem: Combining Query Entity Linking and Entity based Document Ranking. CIKM (2017)
- [4] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, Tie-Yan Liu. Towards Better Text Understanding and Retrieval through Kernel Entity Saliency Modeling. SIGIR (2018)