Neural Estimation of Term Salience for Information Retrieval

Anonymous Author(s)

Abstract

Neural ranking models, trained on large amounts of data, have shown promising performance in retrieval tasks, by learning word and compositional embeddings for effective query-document matching. Neural ranking models typically only consider query to document term matching, and often neglect an essential component of *classical* Information Retrieval (IR) models, namely the importance or salience of each individual term, normally captured by measures of inverse document or collection frequencies. We revisit the idea of term salience in the context of neural IR, by proposing a neural term salience estimation component, integrated into three recent neural ranking models. The term salience component is a parameterized function of the embedding of each individual term that is used as a gate on term matching scores, and is trained end-to-end with the rest of the model. We evaluate our models on a popular large-scale retrieval collection, showing consistent significant improvements over the original neural models as well as other strong baselines. Finally, we observe that, substituting traditional IDF values by the newly proposed salience measure, we can significantly improve classical IR models.

1 Introduction

Neural IR models offer a powerful approach to query-document relevance estimation, by learning embeddings of words and their compositions. Such models have shown promising performance in retrieval tasks in domains with large amounts of available annotated data.

Recent advances show that the most effective neural IR models are *interaction-focused models* (Pang et al., 2017; Zhang et al., 2016), where first the query-to-document term similarities are calculated, and then the similarity scores are used to estimate the final relevance score (Dai et al., 2018; Hui et al., 2018; Fan et al., 2018; Xiong et al., 2017; Hui et al., 2017; Pang et al., 2016; Guo et al., 2016; Hu et al., 2014; Lu and Li, 2013). Interaction-focused neural IR models aim to learn embeddings which can effectively capture semantic similarities (and dissimilarities) of terms, used to estimate query-document relevance. We refer to this first step as *(soft) term matching*, which can be seen as a generalization of the exact matching performed in classical IR models, generally defined as a function of term frequency.

Besides term frequency, another essential component of classical IR models is *term salience*, commonly measured by collection statistics, such as Inverse Document Frequency (IDF) (Robertson, 2004). In fact, in a broad perspective, the relevance score estimation of classical IR models can be formulated as a composition of these two main components. One typical formulation of such a composition – as in the BM25 model (Robertson et al., 2009) – is shown in the following:

$$f(q, d) = \sum_{q_i \in q} f^m(q_i, d)\ f^s(q_i) \qquad (1)$$

where q and d denote the set of query and document terms, q_i is the ith query term, f^m stands for a function of the number (frequency) of exact matches of a term in a document, and f^s estimates the salience of the given term.

We argue that previously proposed neural ranking models have mainly focused on the estimation of f^m, the function based on the interactions between query to document terms, while modeling f^s, the salience function defined over individual terms, has been largely neglected.

In this work, we revisit the idea of term salience from the perspective of neural IR models, proposing a novel neural component to capture term salience. Our proposed approach is generic, and can be integrated into any neural IR model. In

particular, we study extending three recent neural models with our proposed neural term salience component, namely Kernel-based Neural Ranking Model (KNRM) (Xiong et al., 2017) Convolutional KNRM (ConvKNRM) (Dai et al., 2018), and Position-Aware Convolutional-Recurrent Relevance Matching (PACRR) (Hui et al., 2017). These models are selected due to their recent strong performance on retrieval tasks as well as their diversity in model architectures. Our novel extensions to these models enable an effective measurement of both term matching and term salience for relevance prediction.

Our neural term salience component is a parameterized function of word/compositional embeddings, and is calculated separately for each query and document term. To integrate this component with the neural IR models, the salience scores of a query and a document term are applied as a gate mechanism to their term matching result. In this way, the neural term salience component controls the information flow of the term matching based on the salience of both document and query terms. The neural term salience component is learned end-to-end with the rest of the original models.

We evaluate our models enhanced with the salience component on the MS MARCO Passage Reranking (Nguyen et al., 2016) collection, a recent large-scale retrieval dataset in the Web domain, with the Mean Reciprocal Rank (MRR) (Voorhees, 1999) and Recall measures. Our experiments show significant improvements of our approach over all the original models, as well as in comparison to strong baselines.

In addition, we analyze the terms' salience scores calculated by the neural salience components of the trained ranking models. We compare the salience scores with IDF values of the terms, calculated based on collection statistics. We observe considerable correlations between the IDF values and the salience scores, as well as high similarities between their distributions, where the salience scores show a notably smoother distribution. We also analyze the neural salience scores by using them to replace IDF values in the well-known BM25 model, and observe significant improvement over the original BM25 model.

Our novel contribution is three-fold:

- Introducing a novel neural method for capturing term salience
- Applying our proposed term salience method to three recent neural IR models
- Demonstrating significant improvements over state-of-the-art neural IR models on a recent large-scale collection

The remainder of this paper is structured as follows: We review the related work and explain the three neural ranking models in Section 2. We introduce our term salience models in Section 3. The experimental design is described in Section 4, followed by a presentation of results in Section 5. Finally, in Section 6, we discuss our analysis of the neural salience scores.

2 Background and Related Work

2.1 Related Studies

Within the family of interaction-focused neural ranking models, all models start by computing similarities between query terms and document terms. We can then distinguish between two groups of models. The first group captures patterns of similarity values across terms that are close together within the query and within the document (Fan et al., 2018; Hui et al., 2018, 2017; Pang et al., 2016). The second group captures patterns of frequencies across ranges of similarity values (Dai et al., 2018; Xiong et al., 2017). In this work, we integrate our neural salience approach into one model from the former group, and two models from the latter group.

Related to our study, previous work demonstrates the effect of exploiting term salience with ranking models (Hui et al., 2018; Guo et al., 2016; Pang et al., 2016) by directly using pre-computed IDF scores in the model as an external factor. In particular, and as a motivating example to this work, (Galkó and Eickhoff, 2018) note considerable performance difference in response to different salience estimation schemes. Other studies exploit term similarities of pre-trained word embeddings to extend the term matching and term salience components of classical IR models (Rekabsaz et al., 2016, 2017a). In this direction, recent work highlights the importance of adapting word embeddings for IR by post-filtering (Rekabsaz et al., 2017b), retrofitting (Hofstätter et al., 2019) and re-training on local documents (Diaz et al., 2016). In contrast to these studies, our neural term salience component is integrated in the neural models, and trained end-to-end with the model.

2.2 Neural Ranking Models

Given query q with n_q terms and document d with n_d terms, neural IR models estimate the relevance of the document to the query with a function f, typically optimized using a pairwise objective, formalized as follows:

$$\mathcal{L} = \sum_{q} \sum_{d^+, d^- \in D_q^{+-}} \max(0, 1 - f(q, d^+) + f(q, d^-))$$

where d^+ and d^- refer to a relevant and non-relevant document to the query from the set of training pairs $D_q^{+,-}$.

To estimate the relevance function f, the neural ranking models first map all terms to word embedding space. This results in the matrices of query embeddings $\boldsymbol{Q}$ and document embeddings $\boldsymbol{D}$, with dimensions $n_q \times n_v$ and $n_d \times n_v$ respectively, where n_v is the size of the embedding vectors. We assume that the rows of $\boldsymbol{Q}$ and $\boldsymbol{D}$ are in the same order as the terms in the query and document, respectively. In the following, we describe the key points of the KNRM, ConvKNRM, and PACRR models. The architectures of the three models are depicted in Figure 1, when considering only the grey and black sections of the schematic.

2.2.1 KNRM

Given the embedding matrices, the KNRM model creates its matching matrix $\boldsymbol{M}$ as follows:

$$M_{i,j} = \text{cosine}(\boldsymbol{q}_i, \boldsymbol{d}_j) \tag{2}$$

where $\boldsymbol{q}_i$ and $\boldsymbol{d}_j$ are the ith and jth embedding vectors in $\boldsymbol{Q}$ and $\boldsymbol{D}$, respectively. In the next step, given n_k (unnormalized) Gaussian kernels g^k, each with mean μ_k and standard deviation σ_k, the KNRM model calculates the kernel output of every row (query) of the matrix $\boldsymbol{M}$, as follows:

$$g_i^k(\boldsymbol{M}) = \sum_j \exp(-\frac{(M_{i,j} - \mu_k)^2}{2\sigma_k^2})$$

The kernel outputs of g^k are then calculated for all query terms, and summed up: $\phi^k(\boldsymbol{M}) = \sum_i \log g_i^k(\boldsymbol{M})$. By repeating this process for all kernels, the model creates the vector $\boldsymbol{\phi}$ in n_k dimensions, which is finally used in a feed-forward layer (with $\boldsymbol{w}^o$ and b^o as parameters) to estimate the relevance score:

$$f(q, d) = \tanh(\boldsymbol{w}^{o\top} \boldsymbol{\phi}(\boldsymbol{M}) + b^o) \tag{3}$$

2.2.2 ConvKNRM

The ConvKNRM model extends the KNRM model by adding a Convolutional Neural Network (CNN) (LeCun et al., 2015) layer on top of the term embeddings. Given a convolution vector of size $h \in \{1, ..., n_h\}$, where n_h denotes the maximum filter size, the compositional vectors $\boldsymbol{u}_i$ and $\boldsymbol{e}_j$ are created from the query and document matrix embeddings, respectively:

$$\boldsymbol{u}_i^h = \text{relu}\left(\boldsymbol{W}^{h\top} \boldsymbol{Q}_{i:i+h} + \boldsymbol{b}^h\right),$$
$$\boldsymbol{e}_j^h = \text{relu}\left(\boldsymbol{W}^{h\top} \boldsymbol{D}_{j:j+h} + \boldsymbol{b}^h\right)$$

where $\boldsymbol{W}^h$ and $\boldsymbol{b}^h$ are the CNN parameters for the convolution filter of size h, defined in dimensions $(h \cdot n_v) \times n_u$ and $1 \times n_u$ dimensions, respectively (n_u the output dimension of the CNN layer). $\boldsymbol{Q}_{i:i+h}$ refers to the query embeddings starting from the ith term and spanning the following h terms. Based on the filter size h, the compositional vectors represent h-gram embeddings of the terms.

Similar to the KRNM model in Eq. 2, the query and document compositional vectors created with h and h' convolution filter sizes, are then used to calculate the matching matrix $\boldsymbol{M}^{h,h'}$, as follows:

$$M_{i,j}^{h,h'} = \text{cosine}(\boldsymbol{u}_i^h, \boldsymbol{e}_j^{h'}) \tag{4}$$

Such match matrices are calculated from all combinations of query compositional embeddings to the ones of the document (n_h^2 matrices), and stored in the collection $\mathbb{M}$. Next, the kernel functions are applied on each matching matrix, and their output vectors are concatenated. This results in a vector of size $n_k n_h^2$, which is finally passed through a feed-forward layer, as shown in Eq. 3, to calculate the final relevance score.

2.2.3 PACRR

The PACRR model, similar to KNRM and ConvKNRM, exploits the term matching matrix $\boldsymbol{M}$ to estimate query-document relevance. However, particular to this model, PACRR takes advantage of the fact that entries that are close together in $\boldsymbol{M}$ are close together in the query and document strings. The PACRR model then captures patterns of proximities of similarity values in the matching matrix with a set of CNN layers.

Given the matching matrix $\boldsymbol{M}$ in Eq. 2, the model applies n_l parallel 2-D CNN layers with filter sizes of 1×1, 2×2, ...,$n_l \times n_l$, respectively,

each with n_f filters. Every CNN layer therefore outputs n_f matrices of size $n_q \times n_d$. These n_f convolved matrices for the lth CNN layer are then aggregated to one $n_q \times n_d$ matrix using a max pooling layer. We refer to this final matrix as $\boldsymbol{C}^{(l)}$[1]. Let us summarize the described step with the formula: $\boldsymbol{C}^{(l)} = \texttt{CNN}^m_{l\times l}(\boldsymbol{M})$, where $\texttt{CNN}^m_{l\times l}$ refers to the convolution layer for the matching matrix with filter sizes of $l \times l$ as well as the max pooling function over the convolved matrices of the filters.

The model then applies k-max pooling with k equal to n_p, on each row (query) of the $\boldsymbol{C}^{(l)}$ matrix to select the most important features, shown as follows:

$$\boldsymbol{p}_i^{(l)} = \texttt{MP}_{n_p}(\boldsymbol{c}_i^{(l)}) \tag{5}$$

where $\texttt{MP}_{n_p}$ refers to the k-max pooling function with k set to n_p, and $\boldsymbol{c}_i^{(l)}$ is the vector of the ith row of $\boldsymbol{C}^{(l)}$ (related to the ith query term). The $\boldsymbol{p}_i^{(l)}$ vector has n_p dimensions.

Finally, the $\boldsymbol{p}_i^{(l)}$ vectors of all CNN layers and query terms are concatenated, and used as features for a non-linear function — such as a LSTM or a multi-layer feed-forward network (Hui et al., 2018) — to estimate the relevance score. It is these features $\boldsymbol{p}_i^{(l)}$ which represent the term matching aspect of the model. To also include term salience, the PACRR model adds the IDF values of the query terms as extra features to the term matching features, where the IDF values are normalized with the softmax function.

3 Neural Models with Term Salience

The KNRM, ConvKNRM, and PACRR models propose end-to-end approaches to training embeddings for effective term matching — to capture semantic similarity — and eventually relevance estimation. In this section, we first introduce our neural term salience component, a generic neural architecture for capturing term salience as a complement to the term matching component of neural IR models. We then explain our method for integrating the salience component with each neural ranking model.

3.1 Neural Term Salience

In classical IR models, term salience is a function of the statistics of each individual term in the background collection. In our proposed method, it is a function of the embedding of each term and the training data, indicating the overall importance of the term. To define such a salience function for term t, we first obtain an unnormalized salience value from the term's embedding vector $\boldsymbol{v}_t$, by applying a multi-layered perceptron, namely:

$$\gamma(\boldsymbol{v}_t) = \boldsymbol{w}^{(1)} \cdot \tanh(\boldsymbol{W}^{(2)} \cdot {\boldsymbol{v}_t}^\top) \tag{6}$$

where $\boldsymbol{w}^{(1)}$ and $\boldsymbol{W}^{(2)}$ are the model parameters, with sizes $1 \times n_s$ and $n_s \times n_v$, given n_s as the size of the projection vectors, and $\tanh$ is the hyperbolic tangent function.

The result of the γ function is unnormalized, and directly combining it with term matching scores can cause unexpected behaviour during training, as one component may dominate the effect of the other. To avoid this, we define term salience as a gate mechanism that controls the information flow of term matching scores. We therefore squash the output of γ with the sigmoid function to the $[0, 1]$ range, giving rise to the following term salience function:

$$\text{salc}(v_t) = \text{sigmoid}(\gamma(\boldsymbol{v}_t)) \tag{7}$$

The $\text{salc}(v_t)$ value can then be used together with the term matching scores of arbitrary neural models, as demonstrated in the following.

3.2 Integration with Ranking Models

In the following, we explain our approach to extending the neural ranking models discussed in Section 2 with our salience component. We refer to the novel models as *Salience-enhanced KNRM (S-KNRM)*, *Salience-enhanced ConvKNRM (S-ConvKNRM)*, and *Salience-enhanced PACRR (S-PACRR)*. The architectures of our proposed models are illustrated in Figure 1.

3.2.1 S-KNRM

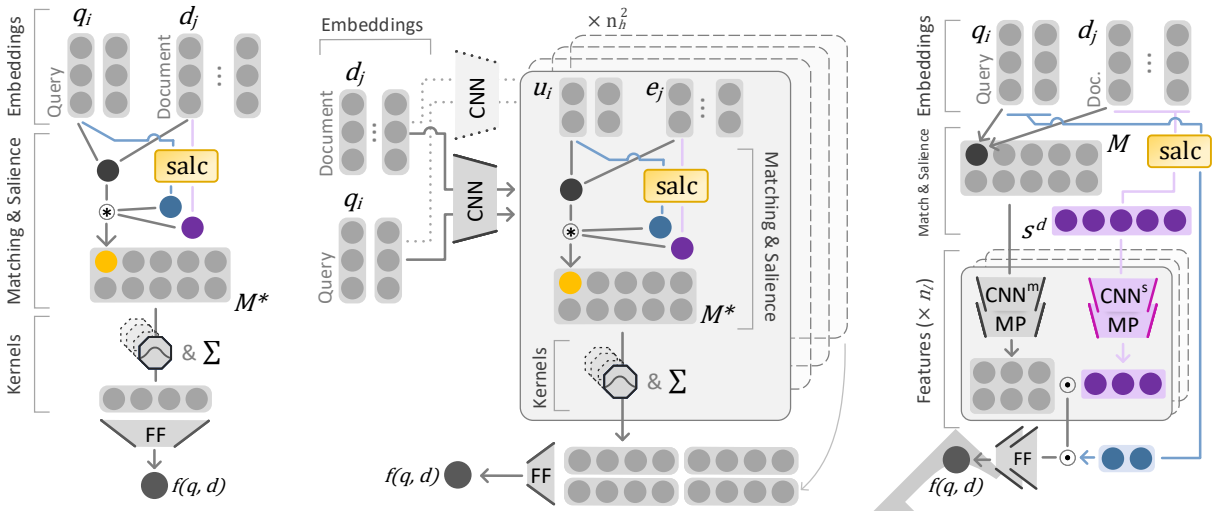
To integrate the term salience component with the KNRM model, we first calculate the salience function for the i-th query term and the j-th document term using Eq. 7, shown as follows:

$$\text{salc}(q_i) = \text{sigmoid}(\gamma(\boldsymbol{q}_i)) \tag{8}$$

$$\text{salc}(d_j) = \text{sigmoid}(\gamma(\boldsymbol{d}_j)) \tag{9}$$

To combine $\text{salc}(q_i)$ and $\text{salc}(d_j)$ with the KNRM model, we multiply these salience values with the term matching value of the query and document terms, defined in Eq. 2, and finally pass them

[1]For $l = 1$, the CNN layer is skipped, and the original matching matrix is replaced: $\boldsymbol{C}^{(1)} = \boldsymbol{M}$



(a) S-KNRM (b) S-ConvKNRM (c) S-PACRR

Figure 1: Architectures of the ranking models, enhanced with the neural term salience component. The elements of the original neural ranking models, namely KNRM, ConvKNRM, and PACRR, are depicted in grey and black, and the novel contributions are shown in color

through the hyperbolic tangent function. This results in a revised definition of matrix $\boldsymbol{M}$, referred to as $\boldsymbol{M}^*$, formulated as follows:

$$M^*_{i,j} = \tanh\big(M_{i,j}\ \text{salc}(q_i)\ \text{salc}(d_j)\big) \quad (10)$$

Using the tanh function, the values of $\boldsymbol{M}^*$ remain in the $[-1, 1]$ range that is required by the kernel functions in the next steps of the model[2]. A $M^*_{i,j}$ value will be high if the terms q_i and d_j are similar and they both have high salience scores. The rest of the S-KNRM model remains the same as the baseline KNRM.

3.2.2 S-ConvKNRM

Integrating salience with the ConvKNRM model follows a similar approach. Instead of using word embeddings, the salience function is calculated on the compositional embeddings resulting from the CNN layer. The salience function for the query and document embeddings with convolutional filter h and h' respectively, is defined as follows:

$$\text{salc}(u^h_i) = \text{sigmoid}\big(\gamma(\boldsymbol{u}^h_i)\big) \quad (11)$$

$$\text{salc}(e^{h'}_j) = \text{sigmoid}\big(\gamma(\boldsymbol{e}^{h'}_j)\big) \quad (12)$$

Similar to S-KNRM, using the salience values, the matrix $\boldsymbol{M}^{h,h'}$ in Eq. 4 is re-defined as $\boldsymbol{M}^{*^{h,h'}}$, formulated as follows:

$$M^{*^{h,h'}}_{i,j} = \tanh\big(M^{h,h'}_{i,j}\ \text{salc}(u^h_i)\ \text{salc}(e^{h'}_j)\big) \quad (13)$$

[2]We should note that even without tanh the values remain in the $[-1, 1]$ range. However, in preliminary experiments, we observed better performance when using the tanh function in Eq. 10.

The rest of the S-ConvKNRM model remains unchanged from the ConvKNRM.

As can be seen from the equations, the S-ConvKNRM model uses the same salience component (the same set of parameters) for all n-gram embeddings. This is a reasonable design decision since, based on the architecture of ConvKNRM, the compositional embeddings are all situated in the same representation space, and therefore can all share the same set of salience parameters.

3.2.3 S-PACRR

In the KNRM and ConvKNRM models, the salience scores of query and document terms are directly multiplied to the match matrices of the models. However, as observed in our pilot experiments, such an approach is not as effective for the PACRR model, resulting in limited performance gains over the original model. We argue that this happens due to the conceptual difference of PACRR to KNRM and ConvKNRM, where a set of CNN layers are applied *after* calculating the matching matrix. Therefore, by directly applying the salience scores to the matching matrix, the CNN layers do not have the possibility to distinguish between the term matching and term salience signals.

To address this issue, we define neural term salience components whose output scores gate the feature vectors in Eq. 5. Looking at that equation, the feature vector $\boldsymbol{p}^l_i$ is specified for every query term, but it is aggregated across document terms. We therefore calculate the salience of query terms

and document terms separately, as explained in the following.

For the document terms' salience scores, we first define the vector $\boldsymbol{s}^d$ with n_d dimensions, containing the salience scores of all document terms, calculated using the salc function as follows:

$$s_j^d = \text{salc}(d_j) = \text{sigmoid}(\gamma(\boldsymbol{d}_j)) \qquad (14)$$

where s_j^d refers to the jth element of the $\boldsymbol{s}^d$ vector.

Analogously to the PACRR model, we then apply a set of CNN layers followed by k-max pooling, to compute a feature vector for the document salience scores. First, we pass the $\boldsymbol{s}^d$ vector to n_l parallel 1-D CNN layers with filter sizes of 1, 2, ...,n_l, respectively, and each with n_f filters. Every CNN layer therefore outputs n_f vectors of size n_d. The n_f convolved vectors of the lth CNN layer are then aggregated using a max pooling layer, resulting to one vector with n_d dimensions. We refer to these steps of the CNN layer and max pooling as $\texttt{CNN}_l^s$.

Finally, a k-max pooling layer with k equal to n_p is applied, resulting in a vector of n_p dimensions. We refer to this vector as the salience feature vector of the document terms, denoted by $\boldsymbol{s}^{(l)}$. The following formula summarizes the mentioned steps: $\boldsymbol{s}^{(l)} = \texttt{MP}_{n_p}\left(\texttt{CNN}_l^s(\boldsymbol{s}^d)\right)$

For query terms, we calculate the salience score of each term, using the salc function, shown in the following:

$$\text{salc}(q_i) = \text{sigmoid}(\gamma(\boldsymbol{q}_i)) \qquad (15)$$

Given the salience feature vector of document terms $\boldsymbol{s}^{(l)}$ and the salience value of the ith query term $\text{salc}(q_i)$, we redefine the feature vector $\boldsymbol{p}_i^{(l)}$ (Eq. 5) as $\boldsymbol{p}_i^{*(l)}$, as follows:

$$\boldsymbol{p}_i^{*(l)} = \left(\boldsymbol{p}_i^{(l)} \odot \boldsymbol{s}^{(l)}\right) \text{salc}(q_i) \qquad (16)$$

where $\odot$ denotes the element-wise multiplication. The rest of the S-PACRR model is the same as the corresponding baseline PACRR: the new feature vectors are concatenated and passed to a non-linear function for relevance score estimation. The IDF features are not used in S-PACRR as the salience is already integrated in the model.

4 Experiment Design

Collection We conduct our experiments on the MS MARCO (Nguyen et al., 2016) Passage Re-ranking collection. The collection provides a large set of informational question-style queries from Bing's search logs, accompanied by human-annotated relevant/non-relevant passages. The collection consists of 8,841,822 documents, around 40 million training data points, and 55,578 queries in the development set.

Baselines *Exact Matching Models:* we use the Query Likelihood Language Modeling (LM) (Lavrenko and Croft, 2001) with Dirichlet smoothing, BM25, and an RM3 Psuedo Relevance Feedback (PRF) model (Lv and Zhai, 2009; Lavrenko and Croft, 2001) as strong exact matching and query expansion IR baselines.

MatchPyramid: In this model (Pang et al., 2016), inspired by neural computer vision models, a set of hierarchical convolution layers is applied on top of the term matching matrix to estimate the relevance score.

Original Neural Models: KNRM, ConvKNRM and PACRR

KNRM-withIDF: a version of KNRM, enhanced with IDF scores. The model is similar to S-KNRM, while the salience scores are replaced by IDF scores, normalized between 0 and 1

PACRR-noIDF: a version of PACRR without the IDF features

We should note that combining IDF with ConvKNRM in a reasonable way is not possible, as IDF is only provided for unigram terms.

Evaluation Performance evaluation is conducted in terms of the Mean Reciprocal Rank (Voorhees, 1999) measure with a cut-off of 10 (MRR@10), and Recall@10. Statistical significance tests are done using a two sided paired t-test and significance is reported for $p < 0.01$.

The details of collection preparation, preprocessing, model parameters, training, and implementation are described in Appendix A (supplementary materials)[3].

5 Evaluation Results

The evaluation results of the models on the validation and test sets of the MS MARCO collection for the MRR and Recall metrics are shown in Table 1. All model variants based on the same original model/category are grouped together, separated by dashed lines. The best results in each group are shown in bold, and the best results overall are indicated with underlines. For brevity, we

[3] Our code is available in supplementary materials and will be published with the paper

Table 1: Evaluation results on the MS MARCO collection. The best results in each group are indicated by bold numbers, and the overall best with underlines. The signs show the significance improvements ($p < 0.01$) over all the baselines, indicated with the signs inside the parentheses: the sign a refers to LM, BM25, and RM3 PRF, b to MatchPyramid, c to both KNRM and KNRM-withIDF, d to PACRR-noIDF and PACRR, and e to ConvKNRM

Model	Validation Set		Test Set	
	MRR	Recall	MRR	Recall
LM (a)	0.186	0.391	0.188	0.388
BM25 (a)	0.190	0.397	0.192	0.398
RM3 PRF (a)	0.192	0.405	0.194	0.405
MatchPyramid (b)	0.234^{ac}	0.443^{a}	0.201^{a}	0.355
KNRM (c)	0.221^{a}	0.439^{a}	0.213^{ab}	0.402^{b}
KNRM-withIDF (c)	0.225^{a}	0.441^{a}	0.217^{ab}	0.428^{ab}
S-KNRM	$\mathbf{0.237}^{ac}$	$\mathbf{0.470}^{abc}$	$\mathbf{0.237}^{abc}$	$\mathbf{0.467}^{abc}$
PACRR-noIDF (d)	0.241^{abc}	0.465^{abc}	0.242^{abc}	0.467^{abc}
PACRR (d)	0.244^{abc}	0.468^{abc}	0.242^{abc}	0.467^{abc}
S-PACRR	$\mathbf{0.253}^{abcd}$	$\mathbf{0.474}^{abc}$	$\mathbf{0.251}^{abcd}$	$\mathbf{0.480}^{abcd}$
ConvKNRM (e)	0.261^{abcd}	0.487^{abcd}	0.254^{abcd}	0.481^{abcd}
S-ConvKNRM	$\underline{\mathbf{0.272}}^{abcde}$	$\underline{\mathbf{0.504}}^{abcde}$	$\underline{\mathbf{0.271}}^{abcde}$	$\underline{\mathbf{0.505}}^{abcde}$

assign the same sign of significance to the baselines with similar results. The sign related to each set of similar baselines indicates significant improvements over all the models in that set.

As shown, in each group of models, the models with the salience component, namely S-KNRM, S-PACRR, and S-ConvKNRM, consistently outperform other models in their groups on both evaluation metrics. The improvements of the models with neural term salience are significant in all cases on the test set. The results confirm the effectiveness of exploiting the neural term salience component for the ranking models.

In general, the neural models outperform the exact matching ones. Between the neural models, ConvKNRM and S-ConvKNRM show the best performance, where S-ConvKNRM significantly improves the previously introduced ranking models in both evaluation metrics.

When inspecting the results of KNRM-withIDF and PACRR, in comparison to KNRM and PACRR-noIDF respectively, we observe only marginal improvements when using the IDF values in the models. In contrast, integrating the models with the neural term salience component in S-KNRM and S-PACRR brings significantly larger improvements to the original models, with or without IDF. These results emphasize the importance of considering term salience in the design of neural ranking models, and in particular the effect of integration of the neural salience component with the models.

In the next section, we continue our investigations by taking a closer look at the differences/similarities between neural term salience scores and IDF values.

6 Analysis of Neural Salience

To compare the term salience scores of the neural models with IDF, we focus on the outputs of the salience components of the S-PACRR, S-KNRM, and S-ConvKNRM models, trained on the MS MARCO collection. We calculate the salience scores of all unigram terms of the collection in each model, using the outputs of the salc function (Eq. 7) given the embedding of each term. In the S-ConvKNRM model, we only use the CNN layer with filter size one, which again provides a unigram term representation. For the same set of unigram terms ($\sim$365K unique terms), we calculate the IDF values, using the statistics of the collection.

Figure 2 shows the histogram of the salience scores of the unigram terms, achieved from S-KNRM and S-ConvKNRM, as well as the histogram of the IDF values (200 bins for all plots). The IDF values are normalized between 0 and 1 using the min-max normalization. As shown, the general shapes of the histograms of the neural term salience scores resemble the histogram of the IDF values, namely skewed toward high values and long tails in the area with lower values. However in contrast to the IDF's histogram, the neural salience scores are distributed more smoothly over

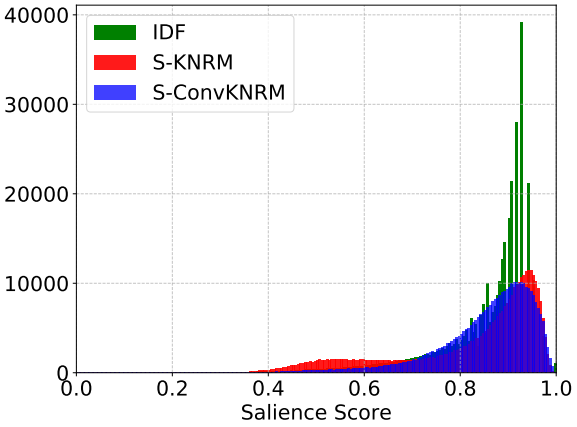


Figure 2: Histogram of the unigram term salience scores, achieved from the salience components of the models, and the IDF scores (scaled between 0 to 1)

the range of salience values, which is an expected behaviour from the outputs of neural networks.

The histogram of the unigram term salience scores of S-PACRR is not shown in the figure since, in contrast to the other models, almost all of its scores are close to one, making the plot very disproportional. We assume that this is due to the architecture of S-PACRR, since the salience component is trained on providing salience features, composed from the term salience scores of document terms. It is therefore expected that extracting unigram term salience scores, out of the context of the model does not provide informative results.

The Spearman Rank correlation coefficients between the neural unigram term salience scores and IDF values are shown in the middle part of Table 2. The results show significant correlations (all have extremely small p values) especially for the scores of S-KNRM and S-ConvKNRM, while S-ConvKNRM is slightly higher than S-KNRM.

We continue our analysis by investigating the effect of using the unigram term salience scores, when plugged in as the salience component in a exact matching model, namely BM25. We do so by replacing the IDF values in the BM25 model (the f^s component in Eq. 1) with the term salience scores of the neural models[4].

The evaluation results on the test set of the MS MARCO collection are shown on the right side of Table 2. The first two rows indicate a BM25 model without the f^s component, and the original BM25 model, respectively. The S-PACRR model — as expected — performs

[4]We observe that removing sigmoid marginally improves the results, as it allows for a wider range of values. We therefore use the outputs of the γ function (Eq. 6) instead of salc in these experiments

Table 2: Middle part: Spearman Rank correlation coefficient of the unigram term salience scores to the IDF values. Right part: evaluation results of the BM25 models, where the IDF values are replaced with the salience scores. The † sign denotes significant improvements to the original BM25 model

Salience Score	Correlation to IDF	BM25 Results MRR	Recall
IDF removed	-	0.126	0.242
IDF (no change)	-	0.187	0.383
S-PACRR	0.10	0.113	0.219
S-KNRM	0.22	0.185	0.371
S-ConvKNRM	**0.23**	**0.195**†	**0.396**†

poorly, similar to the BM25 variant without IDF. On the other hand, S-KNRM shows very similar performance to the original BM25, and interestingly S-ConvKNRM significantly outperforms the BM25 model.

These results are in fact quite surprising, considering that the proposed term salience scores are obtained from neural components, trained end-to-end with other parts of the neural ranking models in completely different contexts. At the same time, the f^m and f^s components of BM25 are specifically designed to work together. Improving BM25 by replacing the f^s component with even the unscaled outputs of the neural term salience components of the trained ranking models, supports our hypothesis on the effectiveness of capturing term salience in our proposed neural architecture.

7 Conclusion

This study highlights the importance and effect of term salience in the design of neural ranking models. We propose a neural term salience estimation component, defined as a function of each query/document term, to capture the extent of its importance. We integrate the proposed term salience component into three recent neural ranking models, proposing their novel salience-enhanced variations, referred to as S-KNRM, S-ConvKNRM, and S-PACRR. Evaluation results on the MS MARCO collection show the significant improvements of our proposed model variants over the original ones in terms of both MRR@10 and RECALL@10. We further explore the relationship between unigram term salience scores achieved from the neural ranking models and traditional IDF values, observing considerable correlations, and even improving the traditional BM25 model by replacing its IDF values with the term salience scores of S-ConvKNRM.

References

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*. ACM, pages 126–134.

Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 367–377.

Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. 2018. Modeling diverse relevance patterns in ad-hoc retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, pages 375–384.

Ferenc Galkó and Carsten Eickhoff. 2018. Biomedical question answering via weighted neural network passage retrieval. In *Proceedings of the 40th European Conference on Information Retrieval (ECIR)*. Springer.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pages 55–64.

Sebastian Hofstätter, Navid Rekabsaz, Mihai Lupu, Carsten Eickhoff, and Allan Hanbury. 2019. Enriching word embeddings for patent retrieval with global context. In *In Proceedings of European Conference of Information Retrieval*.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A position-aware neural ir model for relevance matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1049–1058.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2018. Co-PACRR: A context-aware neural IR model for ad-hoc retrieval. In *Proceedings of the eleventh ACM international conference on web search and data mining*. ACM, pages 279–287.

Victor Lavrenko and W Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 120–127.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* .

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*. pages 1367–1375.

Yuanhua Lv and ChengXiang Zhai. 2009. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, pages 1895–1898.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* .

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2017. A deep investigation of deep ir models. *Neu-IR Workshop at the International ACM SIGIR conference on research and development in information retrieval* .

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Navid Rekabsaz, Mihai Lupu, and Allan Hanbury. 2017a. Exploration of a threshold for similarity based on uncertainty in word embedding. In *European Conference on Information Retrieval*. Springer, pages 396–409.

Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Hamed Zamani. 2017b. Word embedding causes topic shifting; exploit global context! In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pages 1105–1108.

Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Guido Zuccon. 2016. Generalizing translation models in the probabilistic relevance framework. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. ACM, pages 711–720.

Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation* .

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval* .

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In *Proceedings of the 8th Text REtrieval Conference (TREC-8)*.

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 International ACM SIGIR conference on research and development in information retrieval*. ACM, pages 55–64.

Ye Zhang, Md Mustafizur Rahman, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, Aaron Angert, Edward Banner, Vivek Khetan, et al. 2016. Neural information retrieval: A literature review. *arXiv preprint arXiv:1611.06792* .