A CONTEXT ALIGNMENT MECHANISM AND RECURRENT NEURAL NETWORKS FOR INFORMATION RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

To overcome the failure of bag-of-words approach in Information Retrieval (IR), recently IR researchers lead toward the use of deep learning models to learn the query and document representations. The relevance score of a document given a query is defined as the distance between the two representations, it can be binary: relevant or not relevant, or on a scale from very relevant to not at all relevant. In this paper, we use the Context-Aligned Recurrent Neural Networks model (CA-RNN) proposed by Qin et al. (2018) to learn low dimension representations of queries and documents. The CA-RNN model is based on a context alignment mechanism and Recurrent Neural Networks (RNN) model. Given a query, documents are ranked using the relevance score. The evaluation results on two datasets extracted from TREC Desk 4 and Desk 5 show a net outperformance of CA-RNN over traditional models like BM25, LSA, BERT and BiLSTM. The context alignment mechanism used in CA-RNN can improve significantly the representations learning to rank IR systems.

1 INTRODUCTION

Information retrieval aims to match a query to relevant documents. Many models were developed to perform IR task, some most popular models are TF-IDF Liu et al. (2018) and BM25 Robertson & Walker (1994) are based on lexical modeling (keyword-matching) where the relevance score is computed from the query and documents terms frequencies. They treat documents as a bag-of-words, words are independent and can not understand polysemy(words with multiple meanings). Also, we find probabilistic modeling is one of the oldest methods used for IR task like query likelihood model (QL) Zhai & Lafferty (2001) where relevance score is computed as the probability of a document is relevant to the query $P(D|Q)$ means the probability of a language model to generate the document given the query. Probabilistic models suffer from the same drawbacks as lexical models, documents are treated as a bag-of-words, words are independent, they ignore the sentence structure, struggle with polysemy and the meaning of words in the context.

To overcome the limitations of the lexical and probabilistic modeling, researchers introduced contextual information modeling to deal effectively with semantic similarity of words rather than identical matching. One of the earliest and most popular semantic model is Latent Semantic Analysis (LSA) Deerwester et al. (1990) which projects query and documents as well as terms into a low dimension space called *latent semantic space*, it maps the query to its relevant documents at semantic level. LSA applies a dimension reduction linear projection technic called Singular Value Decomposition (SVD) to the word/document co-occurrence matrix. Similar documents have similar representations in the latent semantic space even they don't share any term. LSA is able to detect synonyms and polysemy. Many variants of LSA were proposed like Probabilistic LSA (PLSA) Hofmann (1999) that is based on statistics likelihood and can be a generative model of data. These models are trained an unsupervised mode, it works well for classification but not for prediction.

Even LSA model deal well with semantic similarity between documents and terms but it suffer from drawbacks of lexical and probabilistic models. It treats documents as bag-of-words, words are independent and it ignore local phrasing. Recently, application of deep learning Bengio (2009) received more attention and was applied to many domains including information retrieval, Salakhut-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

dinov & Hinton (2007) used deep auto-encoder to develop a sementic hashing model for information retrieval, the model performed better then LSA model.

Deep learning is used to extract hidden features from data and learn representations to perform tasks, particularly IR task. When deep learning applied to IR task, it is formulized as matching problem between the query and documents. Two types of deep learning architectures were proposed to solve matching problem: *representation-based* and *interaction-based.* Representation-based extracts a high level representations of the query and document using a deep learning model then perform the matching between the representations. However interaction-based model, first calculate local interactions of each query and document term at input level, then uses a deep learning model to learn the matching patterns from input interaction matrix.

Many representation-based deep learning models were proposed for web search like DSSM Huang et al. (2013), CDSSM Gao et al. (2010), BLTM and DPM Geo et al. (2011), ARC-I Hu et al. (2013). All these models use feed forward neural or convolutional neural architectures to learn the query and document representations, were trained on very large click-through logs datasets. Authors reported that these models outperformed traditional models like BM25, LSA and PLSA for web search.

Nie et al. (2018) have conducted an empirical study to compare the two architectures representation-based and interaction-based for IR under the same condition. They found interaction-based models perform better than representation-based models. Guo et al. (2016) presented a deep relevance matching model (DRMM) that is interaction-based model and performed better than traditional retrieval models like QL and BM25 on TREC collections that range from small news dataset to large web collection dataset. Authors reported that the representation-based models DSSM Huang et al. (2013), CDSSM Gao et al. (2010) and ARC-I Hu et al. (2013) performed worse then traditional model QL and BM25 for Ad-doc retrieval. These models were evaluated also on $<$ query, doc title $>$ pairs as in the original paper using TREC collections datasets used for DRMM Guo et al. (2016) and performed worse than QL, BM25 and DRMM Guo et al. (2016).

Convolutional Neural Network CNN and Deep Neural Network DNN are the most preferred architectures used to learn the query and document representations for IR task. Recurrent neural network (RNN) models are less preferred for long text because of the gradient vanishes for long sentences and can not capture the essential features. RNN architecture for sentences similarity was introduced by Mueller & Thyagarajan (2016) to compute similarity score of variable-length sentences. It was used by Yang et al. (2016) for semantic matching of questions and answers, and showed RNN perform well for ranking short answer text.

In this paper we use the Context-Aligned Recurrent Neural Networks model (CA-RNN) proposed by Qin et al. (2018) to learn representations of queries and documents for IR task. The model is trained in a supervised manner. CA-RNN is a representation-based model, it learns the representations of a query and document using RNN and a context alignment mechanism. The alignment mechanism aligns overlap or semantic terms in the query and document at term level, then uses a context alignment gating to capture the internal interactions of the aligned terms. The model showed tremendous improvement for sentence similarity score on QA and paraphrase identification tasks Qin et al. (2018),

We study how the Context-Aligned Recurrent Neural Networks model will perform on IR task, where the relevance of a document given a query is computed as the distance between the learned representation of the query and document. To show the impact of the context alignment mechanism on the representations learning, then on the relevance score for IR task, we compare the results of CA-RNN with the basic RNN model. The evaluation of the CA-RNN model on two datasets extracted from TREC Desks 4 and 5 show CA-RNN outperform the basic RNN and baseline models BM25, LSA and BERT.

The rest of the paper is organized as follow. Section 2 reviews the related work. In section 3 and 4, we present CA-RNN and BiLSTM (a variant of RNN) models in details. Section 5 presents how the relevance of a document given a query is computed using the representations of the query and document. Section 6 gives details on how models are trained. In section 7, we describe the datasets used in this work and conduct a comparison analysis of CA-RNN for IR with a set of benchmark models. Finally, in the last section we give a conclusion of our work and some ideas for future research.

2 RELATED WORK

In this section we review the deep learning approaches developed for IR. Salakhutdinov & Hinton (2007) presented a deep autoencoder network to learn the word-count vector of documents. The deepest layer was forced to use a small binary variables (32) in order to map documents to memory addresses in such a way that semantically similar documents are located at nearby addresses. Similar documents to a query are grouped in a neighborhood of the query address and can be accessed easily. The model was learned in two stages. In the first stage called "*pre-training*", they used Restricted Boltzmann Machine to learn a deep generative model to map a term vector representation of a document to a low dimensional semantic vector. In the second stage called "*after pretraining*", the weights (model parameters) are fine-tuned to minimize the error between the original term vector of the document and the reconstructed term vector. Their approach is an unsupervised learning model can not be used to differentiate relevant from not relevant documents for a given query. The relevance of a document to a query is based on whether the document and query have the same class label or not. TF-IDF model performed better than their model when retrieving a few top documents, but it outperformed the LSA model.

Huang et al. (2013) presented a representation-based model (DSSM) that uses a deep feed forward neural networks to learn query and documents representations where the relevance of a document given a query is computed as the distance between the two representations. DSSM can be seen as a nonlinear extension of LSA because it performs a non linear mapping of query and documents into a low dimension semantic space. Authors trained the model on clickthrough logs using <query, doc title>. They reported that their model performed better than the existing models like BM25 and LSA. Shen et al. (2014) presented a similar model to DSSM called CDSSM, the model is deep convolutional networks based. CDSSM model also was trained on clickthrough logs using <query, doc title>, and it performed batter than many other models like DSSM, BM25, unigram language model (ULM) and word-based translation model (WTM). Guo et al. (2016) and Pang et al. (2016) showed DSSM and CDSSM perform much less than BM25 for ad-hoc retrieval.

Nie et al. (2018) conducted an experimental study to compare the performance of representation-based and interaction-based category models for IR task using convolutional networks schemas. Models were trained and evaluated on the same training and testing datasets. Authors reported that interaction-based category models perform better than representation-based category models for IR task this is due to their capacity to learn the local interaction patterns rather than learning global representations.

Guo et al. (2016) proposed a deep relevance matching model (DRMM) for ad-hoc retrieval. DRMM employs a joint deep architecture at the query term level for relevance matching. By using matching histogram mapping, a feed forward matching network, and a term gating network, the model can effectively deal with the three relevance matching factors that are: exact matching signals, query term importance, and diverse matching. Experimental results on two representative datasets show that the model can significantly outperform some well-known retrieval model like BM25, QL and ARC-I.

The pre-trained models such BERT Devlin et al. (2019) and GPT Radford et al. (2018) have shown a great effectiveness on many tasks. They are trained on unlabeled large corpus data over different pre-training tasks and they can be fine-tuned with just one additional output layer to perform a specific task. Lu et al. (2020) presented TwinBert model that is BERT based, it learns query and document representations independently. The documents presentations can pre-computed offline and cached in memory which allowed to improve the computing time enormously. Khattab & Zaharia (2020) presented ColBERT model that also is BERT based and learns representations of the query and document separately, it uses late interaction to model their fine-grained similarity using a softmax function. The model computationally is very effective.

3 CONTEXT-ALIGNED RECURRENT NEURAL NETWORKS MODEL OVERVIEW

The Context-Aligned Recurrent Neural Networks model (CA-RNN) was proposed by Qin et al. (2018) for sentence similarity modeling, it outperformed on QA and paraphrase identification tasks. The model employs the Recurrent Neural Networks(RNN) model and a joint context alignment

gate. First, at query and document terms level, a list of pairs of aligned terms (words) in the query and document is detected and retained. Second, the sequences of the hidden states of the query and document are computed by RNN model. Then in the last step a context alignment gating is used to absorb the contextual information of the aligned words in the query and document to boost their learning representations.

Before we present the CA-RNN model in details, we introduce a short review of RNN model. In this work, we do not use the full text of the document, only the title is extracted. A document is presented only by its title. Given a document D (document's title) and a query Q where

$$D = x_1 x_2 x_3 .... x_m$$

and

$$Q = y_1 y_2 y_3 .... y_n$$

x_i and y_j are the sequence of terms (words) of D and Q respectively. We denote by $\mathbf{x_i} \in \mathbb{R}^d$ and $\mathbf{y_j} \in \mathbb{R}^d$ the term embedding of x_i and y_j respectively. RNN model used to model the dependencies in sequential data, at each time step, the current hidden state $\mathbf{h_t^x}$ depends on the curent input $\mathbf{x_t}$ and the previous hidden state $\mathbf{h_{t-1}^x}$ and it is calculated as follow

$$\mathbf{h_t^x} = f(\mathbf{x_t}, \mathbf{h_{t-1}^x})$$

Where f is the RNN model or its variants like long short term memory (LSTM) Hochreiter & Schmidhuber (1997) or GRU Kyunghyun et al. (2014) both of them introduce a memory mechanism to overcome gradient vanishing or exploding. In the rest of the document, we use Bidirectional LSTM model (BiLSTM) as it performs better than most of existing architectures Jozefowicz et al. (2015). The hidden state $\mathbf{h_t^x}$ represents the context information from the first word to the t^{th} word of D. Using $(\mathbf{h_t^x})$ for t=1,...m the sequence of hidden states of D, self-attention mechanism Tan et al. (2016) and maxpooling mechanism Wang & Nyberg (2015), we compute H_D the representation of the document D as shown in the Figure 1.

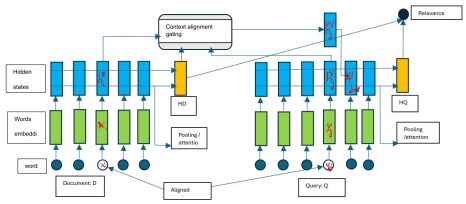


Figure 1: Framework of CA-RNN for Information Retrieval

Below we present The Context-Aligned Recurrent Neural Networks (CA-RNN) model in three steps and the Figure 1 gives a configuration of its framework.

1. **Step 1: Word alignment detecting between document the D and query Q**
   Word alignment detecting consists in finding similar words in D and Q, it can lexical or semantic related words in the pair (D, Q). Two approaches are used to align words: *overlap-based* or *semantic-based*. Overlap-based aims to align lexical overlaps in the pair (D, Q) means the same words that occur in D and Q. Semantic-based approach consists to align words with similar meaning and synonyms in the pair (D, Q). There are many algorithms in literature used to align words in a sentence pair. In this work, we use "*monolingual word aligner algorithm*" developed by Sultan et al. (2015)
2. **Step 2: Compute the hidden states of the terms of D and Q**
   (a) Apply RNN on pre-trained word embeddings of D to compute the sequence of hidden states $(\mathbf{h_t^x})$ for t=1,...m,
   (b) Apply self-attention/maxpooling mechanism on the sequence $(\mathbf{h_t^x})$ to compute H_D the representation of D
   (c) Apply RNN on pre-trained word embeddings of Q to compute the sequence of hidden states $(\mathbf{h_t^y})$ for t=1,...n.

3. **Step 3: Context Alignment Gating**
RNN models the document D and query Q independently and the hidden states contain the contextual information from the first word to the current one. Under the assumption, in a relevant pair (D, Q) the aligned words have similar context. The Context-Aligned Recurrent Neural Networks model can capture the contextual interactions of the aligned words and enhance the relevance of the pair (D, Q) . The framework of the context alignment gating is shown in the Figure 2, it uses the hidden states of the aligned words h_t^x, h_t^y and the document representation H_D to generate a new hidden state h'^y_t as showed below in the equation (2). The context alignment gating mechanism performs two mean steps

   (a) **Relevance measurement:** given h_t^x and h_t^y are the hidden states of the aligned words and H_D is the representation of the document D, the relevance between H_D and h_t^y is formulated as

$$g = \sigma(W_H H_D + W_h h_t^y + b) \tag{1}$$

   where W_H and W_h are weight matrices, b is a bias vector, and $\sigma()$ is an element-wise sigmoid function. The vector g is the relevance in each hidden dimension.

   (b) **Context absorption:** To reduce the contextual gape between the hidden states h_t^x and h_t^y in a relevant pair (D, Q) and increase it in an irrelevant one, we perform the context absorption step to compute the new hidden state h'^y_t as follow

$$h'^y_t = g \odot h_t^x + (1 - g) \odot h_t^y \tag{2}$$

   the new hidden state h'^y_t is passed to the next time step of the sequence of hidden states of the query Q as shown in the Figure 1. The representation of the query Q is generated based on the new sequence of hidden states (h'^y_t) using self-attention/maxpooling mechanism. The context absorption decreases the distance between H_D and H_Q in a relevant pair and increase it in the irrelevant one.

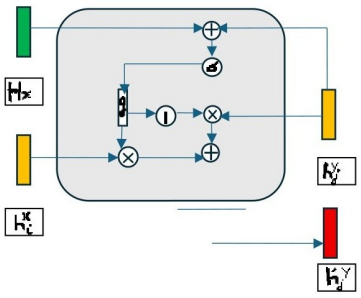


Figure 2: Framework of the Context Alignment Gating.

4 BiLSTM for Information Retrieval

In this section, we present the BiLSTM model that learns the query and document representations separately. More specifically, no aligned gating used to connect the contextual of the document with that of the query as in CA-RNN. The Figure 3 shows a framework of the BiLSTM model for information retrieval. The representations of the document D and query Q are computed from the sequences of the hidden states of D and Q using self-attention/maxpooling mechanism, there is no interaction between the context of the hidden states of the aligned words in D and Q.

In order to study the impact of the alignment mechanism on the learning representations of the document D and query Q, we compare the performance of the model BiLSTM versus the performance of the model CA-RNN. As explained in the section 3, because of the context absorption, CA-RNN can increase the relevance score of a relevant pair (D, Q) and decrease that of an irrelevant one.

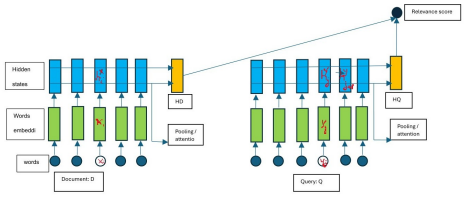


Figure 3: Framework of BiLSTM for Information Retrieval.

5 RELEVANCE SCORING

In this section, we illustrate how to compute the relevance of a document given a query using the models CA-RNN or BiLSTM. A document is presented by its title field, we do not use its description (body text) for representation learning. The query as well as the document are sequences of terms (words). An embedding vector is associated with each term and is learned in an unsupervised way on large corpus. Different architectures may used for learning word embedding like neural networks, recurrent neural networks RNN or convolutional networks CNN.

The relevance score of a document D given a query Q is computed as follow

$$Rel(D|Q) = F \circ \Phi(D, Q) \tag{3}$$

where Φ is the CA-RNN model, it maps the document D and query Q to their representation vectors as explained in the section 3. F is the Manhattan distance function with l_1 norm is defined in the equation (4)

$$F(A, B) = exp(-\|A - B\|_1) \tag{4}$$

We denote by H_D the representation of the document D and H_Q the representation of the query Q

$$\Phi(D, Q) = (H_D, H_Q)$$

and

$$Rel(D|Q) = F(H_D, H_Q) = exp(-\|H_D - H_Q\|_1) \in [0, 1] \tag{5}$$

The distance function F forces the BiLSTM model used in CA-RNN to entirely capture the semantic differences during training as showed by Mueller & Thyagarajan (2016), we point out that using an l_2 rather than l_1 norm in the distance function can lead to up and down in the objective function curve. Chopra et al. (2005) showed during early stages of training, a l_2-based model is unable to correct errors where it erroneously believes semantically different sentences to be nearly identical due to vanishing gradients of the Euclidean distance.

The predicted probability of a Document D given a query Q as 1 is defined by the relevance score $Rel(D|Q)$ as follow.

$$\hat{p}(r = 1|Q, D) = Rel(D|Q)$$

and the predicted probability of a Document D given a query Q as 0 is

$$\hat{p}(r = 0|Q, D) = 1 - Rel(D|Q)$$

When the $\hat{p}(r = 1|Q, D) >= 0.5$ the document is predicted as relevant to the query and labeled as 1 otherwise is labeled 0. It worths to mention that, the relevance of a document D given a query Q as well as its class label is 1 or 0.

In the introduction section, we presented a set of *representation-based* models that build a single representation for the query and document then match the representations using a distance function to compute the relevance. Example includes DSSM Huang et al. (2013), CDSSM Gao et al. (2010) and ARC- I Hu et al. (2013). The model CA-RNN also belong to the category representation-based models, it differs from the models DSSM, CDDSM and ARC-I in two ways:

1. The model CA-RNN uses RNN architecture instead of DNN or CNN architectures, the model RNN can handle sequential data better than DNN and CNN.

2. The model CA-RNN uses a context alignment mechanism that connects the contextual information of the document and query through the aligned terms and that can improve the representations learning to rank IR systems.

6 MODELS TRAINING

To enhance the capture of the past and future context, we use BiLSTM as Φ function (equation 3) for the CA-RNN model. BiLSTM improves the accuracy tasks by aggregating context from both directions. To train the models CA-RNN and BiLSTM for information retrieval, for both, we use a cross-entropy function defined as follow

$$L(Q, D, r, \theta) = -\sum_{j=0}^{1} p(r = j|Q, D)\ log\ \hat{p}(r = j|Q, D)$$

the probability $p(r = 1|Q, D) = 1$ if D is relevant to Q, 0 otherwise, θ includes all the parameters of the neural networks. For optimization, we use stochastic gradient descent (SGD). The initial learning rate is set to 0.01 and momentum set to 0.95. Due to memory constraints, we train our models on two small datasets that are presented in details in the next section. The dimension of the hidden state is set to 128. The architecture for the query and document networks contain the same number of layers, input layer, BiLSTM layer, concatenation layer, self-attention layer, Maxpooling layer to produce a final representation of the input vector and finally the relevance layer to compute the relevance score.

For term embedding, we use 50-dimentional pre-trained Glove(Global Vectors for Word Representation) which are trained on the global word co-occurrence Pennington et al. (2014). To deal with out of vocabulary words, we used "Fast Text Model" to train its embedding with following parameters embedding_size= 50, window_size = 4, min_word = 3, down_sampling = 1e-2, epoch=100 and sg=1. Given the complixity of the models, for a smouth gradient estimate and quick convergence, both models, CA-RNN and BiLSTM were trained for 120 epochs with a batch size of 1024.

The main problem we encountered in training CA-RNN and BiLSTM is the overfitting. in less than 5 epochs, models achieve very good accuracy and very low training error, but it fails to generalize on testing data. The accuracy deteriorate as the number of epochs increase and the testing error gets much large than training error. To overcome the issue of overfitting, we set *dropout* to 0.2 to randomly deactivate a subset of neurons during training, also we used L_1 regularization (Lasso) to apply a penalty on the layer's kernel with argument 0.01.

7 EXPERIMENTS

In this section we describe the datasets used for experiments, baseline models used as benchmark and present the results of the retrieval models.

7.1 DATASETS

To train and test CA-RNN and BiLSTM models, we used TREC *Desk 4 and 5* dataset to build a large dataset contains 198342 rows, where each row is a thrice $<$ Query, Document title, r $>$, r is the label of relevance, it takes 1 if the Document is relevant to the Query, 0 else. In the original dataset, a document is composed by a title and a body, in our dataset only title is extracted. As our computer resources are limited we can not train large models with this original dataset. Given the structure of the original dataset, we extracted first 7000 rows to form a dataset named Data1, equally we extracted last 7000 rows to form another dataset named Data2. This selection is made to keep the structure of Data1 and Data2 as the structure of the original dataset as much as possible. For all datasets, rows are organized in blocks as follow

```
Q_1 D_11 r_11
Q_1 D_12 r_12
Q_1 D_13 r_13
...
```

```
Q_2 D_21 r_21
Q_2 D_22 r_22
Q_2 D_23 r_23
...
```

For a dataset of 7000 rows, the CA-RNN model has around 1 million parameters. Data1 has 8 queries and 7000 documents, an avaerage 875 documents per query. likewise Data2 has 5 queries and 7000 documents, an avarege 1400 documents per query. The table 1 presents the statistics of the train and test datasets. Test datasets are sampled randomly from Data1 and Data2 using a portion of 25% of each dataset. The text of queries and titles was cleaned using a python code before training and testing the models. Words were white space tokenized, lowercased and stemmed, stop words, non-printable, punctuation and numbers were removed.

	# of Queries train/test	# of Docs train/test	Avg Docs/Q train/test
Data1	8/8	5250/1750	656/219
Data2	5/5	5250/1750	1050/350

Table 1: Statistics of train and test datasets

7.2 Baseline Models and Evaluation Metrics

We use four baseline models to compare the performance of CA-RNN. The first is BM25 model is one most popular and traditional probabilistic information retrieval model. The second is LSA model, it uses hidden semantic structures to match query and documents for information retrieval based on word co-occurrence. The third model used as a baseline is BERT (Bidirectional Encoder Representations from Transformers) Devlin et al. (2019) is an unsupervised pre-trained language model, it was successfully applied to IR tasks. Lou et al. (2020) presented a TwinBERT model that used BERT for IR. Another baseline model used in our experiments is bidirectional Long Short-Term Memory (BiLSTM) which is a representation-based deep learning model used to learn the representation of the query and document separately then uses a distance function to compute the relevance as explained in the section 4.

We report the results using two metrics. The Normalized Discounted Cumulative Gain (NDCG) at rank 5 (nDCC@5) and at rank 10 (nDCC@10) Jarvelin & Kekalainen (2000) and the second metric is the mean average precision MAP, it is one of optimized metrics of IR models, it rewards the models that return relevant documents and correct ranking early and it summarizes the precision-recall curve. We report MAP@5 and Map@10, because most of users look at the top 5-10 first documents and it is more relevant to evaluate the relevance on the range 5-10 instead of unrestricted MAP. The Table 2 and Table 3 give a summary results for CA-RNN model and all the baseline models used in this work.

7.3 Results and Analysis

All computing in this work are carried out on Ubuntu Linux 20.04 LT, architecture $X86_64$, 33 GB RAM, 80 GB Disk, 8vCPU Intel. We evaluate the CA-RNN model and the baseline models introduced in the previous section on two datasets Data1 and Data2 and the results are summarized in the tables **Table 2** and **Table 3**. For both datasets Data1 and Data2, the performance of CA-RNN is compared to the performance of BM25, LSA, BERT-Base and BiLSTM. The best results obtained for both datasets are showed in bold. From Table 3, on Data2, the model CA-RNN outperforms all the baseline models for all metric with significant margin. From Table 2, on Data1, CA-RNN outperforms or at least at the same level as all models. For the metric MAP@5, The model CA-RNN performed better than BM25, BERT-Base and BiLSTM, except the model LSA slightly exceeds CA-RNN by 0.62%. We made the same observation for the metric MAP@10, the model BiLSTM performed slightly more (0.55%) than CA-RNN. To measure the significant improvement over models, we used paired t-test at 5% significance level.

	MAP@5	nDCC@5	MAP@10	nDCG@10
BM25	0.1375	0.2166	0.1741	0.3297
LSA	**0.3562**	0.4547	0.2645	0.4006
BiLSTM	0.3000	0.4109	**0.3083**	0.4594
BERT	0.2000	0.3615	0.1624	0.3750
CA-RNN	0.3500	**0.5604**	0.3028	**0.5185**

Table 2: Ranking results for all used models using Data1

	MAP@5	nDCC@5	MAP@10	nDCG@10
BM25	0.2466	0.3495	0.2400	0.3517
LSA	0.0833	0.1141	0.1033	0.1719
BiLSTM	0.1800	0.3016	0.1666	0.3700
BERT	0.0800	0.2358	0.0800	0.2931
CA-RNN	**0.3200**	**0.3660**	**0.3800**	**0.4747**

Table 3: Ranking results for all used models using Data2

For both datasets Data1 and Data2, BM25 performed much lower than CA-RNN model. In BM25 model the relevance score of a document given a query is computed using the matched query and docment terms. However, in CA-RNN the relevance score is computed using the learned representations of the query and document and the representations learning is based on contextual meaning. As the CA-RNN model has two main components, the first is RNN that is able to capture the context of the query and document and models effectively the dependencies between terms. The second component is the context alignment gating that aligns overlap (exact terms) and semantic terms in query and document then connects contextual information of the query and document to improve their learned representations. A better representations may reduce the contextual gap between relevant query and document, and increase it in irrelevant pair. The outperformance of the model CA-RNN is derived from the context alignment mechanism. Qin et al. (2018) showed that a semantic-based alignment approach outperforms the overlap-based one because it can identify more terms between query and document.

To study the impact of the context alignment mechanism on representation learning then on relevance score, we introduced the BiLSTM model that is a combination of two BiLSTM networks, each network learns the representation of the query and document separately then use the distance function (4) to compute the relevance score of the pair query-document. The BiLSTM does not use the context alignment mechanism to connect the hidden states of the aligned terms. Therefore, there is no interaction between the hidden states of the aligned terms of the query and document. The results of the Table 2 and Table 3 show the outperforms of the CA-RNN model over BiLSTM model. Hence the importance of the context alignment mechanism on the representation learning in deep learning.

LSA model is a semantic model that maps queries and documents based on word co-occurrence. It groups similar terms and documents based their contextual meaning. In this work, we used the python library " scikit-learn" package to implement LSA model. The results from Table 2 and Table 3 shows the effectiveness of CA-RNN over the semantic model LSA.

We evaluated CA-RNN vs BERT-base (Bidirectional Encoder Representations from Transformers) which is also a deep learning model and uses bidirectional feedforward network and self-attention mechanism. BERT-base is a pre-trained and fine-tuned model, it is pre-trained on large corpus in unsupervised manner to learn sentence representation. In this work we used BERT-base to learn representations of the queries and documents. To product a high quality of query and document representations, we fine-tuned BERT-base for information retrieval task using our datasets Data1 and Data2 to improve its performance. Results are shown in Table 2 and Table 3, CA-RNN outperforms BERT-base.

REFERENCES

Y. Bengio. Learning deep architectures for AI. *Foundumental Trends Machine Learning*, 2, 2009.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively with application to face verification. *In Computer Vision and Pattern Recognition*, (1):539–546, 2005.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. American Society for Information Science,*, 41(6):391–407, 1990.

J. Devlin, M.W Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *In Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

J. Gao, X. He, and J-Y Nie. Clickthrough-based translation models for web search: from word models to phrase models. *In CIKM*, pp. 1139–1148., 2010.

J. Geo, K. Toutanova, J. Gao, and W-T. Yih. Clickthrough based latent semantic models for web search. *In SIGIR.*, 2011.

J. Guo, Y. Fan, Q. Ait, and W. Bruce. A deep relevance matching model for ad-hoc retrieval. *In CIKM.*, 2016.

S. Hochreiter and J. Schmidhuber. Long shortterm memory. *Neural computation*, 9(8):1735–1780, 1997.

T. Hofmann. *Probabilistic latent semantic indexing.* In SIGIR., 1999.

B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. *In NIPS.*, pp. 2042–2050, 2013.

J. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. *In CIKM*, 2013.

K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. *In SIGIR*, pp. 41–48, 2000.

R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. *In ICML*, pp. 2342–2350, 2015.

O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. *In ACM*, 2020.

C. Kyunghyun, M. Bart, G. Caglar, B. Dzmitry, B. Fethi, S. Holger, and B. Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *In EMNLP*, pp. 1724–1734, 2014.

C. Liu, Y. Sheng, Z. Wei, and Y. Yang. Research of text classification based on improved TF-IDF algorithm. *In IRCE*, pp. 218–222, 2018.

W. Lou, J. Jiao, and R. Zhang. TwinBERT: distilling knowledge to twin-structured bert models for e€icient retrieval. *In CIKM*, pp. 2645–2652, 2020.

W. Lu, J. Jiao, and R. Zhang. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval. *In CIKM*, pp. 2645 – 2652, 2020.

J. Mueller and A. Thyagarajan. Siamese recurrent architectures for learning sentence similarity. *In AAAI*, pp. 2786–2792, 2016.

Y. Nie, Y. Li, and J. Nie. Empirical study of multi-level convolution models for IR based on representations and interactions. *In ICTIR-18*, 2018.

L. Pang, Y. Lan, J. Guo, J. Xu, and X. Cheng. A study of matchpyramid models on ad-hoc retrieval. *In CoRR*, pp. 373–374, 2016.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *In EMNLP*, pp. 1532–1543, 2014.

C. Qin, H. Qinmin, J. Huang, and H. Liang. CA-RNN: using context-aligned recurrent neural networks for modeling sentence similarity. *In AAAI*, 2018.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding with unsupervised learning. *Technical Report, OpenAI*, 2018.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. *In SIGIR*, pp. 232–241, 1994.

R. Salakhutdinov and G. Hinton. Semantic hashing. *in Proc. SIGIR Workshop Information Retrieval and Applications of Graphical Models.*, 2007.

Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. *In Proceedings of WWW '14*, 373–374, 2014.

Md. A. Sultan, S. Bethard, and T. Sumner. Sentence similarity from word alignment and semantic vector composition. *In Proceedings of the 9th International Workshop on Semantic Evaluation*, pp. 148–153, 2015.

M. Tan, C. d. Santos, B. Xiang, and B. Zhou. Lstm based deep learning models for non-factoid answer selection. *ICLR*, 2016.

D. Wang and E. Nyberg. A long short-term memory model for answer sentence selection in question answering. *In ACL*, pp. 707–7012, 2015.

L. Yang, Q. Ai, J. Guo, and W. Bruce Croft. Ranking short answer texts with attention-based neural matching model. *In Proceedings of ACM CIKM2016*, pp. 287–296, 2016.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. *In SIGIR*, pp. 334–342, 2001.