Improving Passage Re-Ranking with Word N-Gram Aware Coattention Encoder

Chaitanya Sai Alaparthi and **Manish Shrivastava**
Language Technologies Research Centre (LTRC),
Kohli Centre on Intelligent Systems(KCIS),
International Institute of Information Technology, Hyderabad, India
chaitanyasai.alaparthi@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In text matching applications, coattentions have proved to be highly effective attention mechanisms. Coattention enables the learning to attend based on computing word level affinity scores between two texts. In this paper, we propose two improvements to coattention mechanism in the context of passage ranking (re-ranking). First, we extend the coattention mechanism by applying it across all word n-grams of query and passage. We show that these word n-gram coattentions can capture local context in query and passage to better judge the relevance between them. Second, we further improve the model performance by proposing a query based attention pooling on passage encodings. We evaluate these two methods on MSMARCO passage re-ranking task. The experiment results shows that these two methods resulted in a relative increase of 8.04% in Mean Reciprocal Rank @10 (MRR@10) compared to the naive coattention mechanism. At the time of writing this paper, our methods are the best non transformer model on MS MARCO passage re-ranking task and are competitive to BERT base while only having less than 10% of the parameters.

1 Introduction

Passage ranking (or re-ranking) is a key information retrieval (IR) task in which a model has to rank (or re-rank) set of passages based on how relevant they are to a given query. It is an integral part of conversational search systems, automated question answering systems (QA), etc., The typical answer extraction process in these systems consists of two main phases. The first phase is ranking passages from the collection that most likely contain the answers. The second phase is extracting answers from these passages. The performance of first phase significantly impact the performance of extracting answers and the performance of the overall system. Thus it is important for a QA system to effectively rank passages.

Attention mechanisms have shown tremendous improvements in the deep learning based NLP models (Bahdanau et al., 2014; Wang et al., 2016; Yang et al., 2016; Lu et al., 2016; Vaswani et al., 2017). Attention allows the model to dynamically focus only on certain parts of the input that helps in performing the task at hand effectively. Coattentions (Xiong et al., 2016) are class of attention mechanisms which can be applied on text matching problems. They proved to be highly effective as they enables the learning to attend based on computing word level affinity scores between two texts thus helping in effectively deciding the relevance between them.

This paper builds on previous work on coattention mechanism (Alaparthi, 2019) (we call it as naive coattention encoder) to tackle the problem of passage re-ranking. We recall that the coattention encoder attends across query and passage by computing the *word-level* affinity scores. Similar to (Hui et al., 2018), we argue that attending at *word-level* limits the ability to capture local context in the query-passage interactions. As an example (which we later explain in section 5.3), for a query: *what is January birthstone color*, the naive coattention encoder can relate the passages describing passages such as *November birthstone color*, *April birthstone color*, etc. This is likely because of common matching terms *birthstone* and *color* and semantically similar words *January*, *November*, *April*, etc. We demonstrate that extending the coattentions to words and n-grams can improve the matching signals, which will contribute to final relevance scores.

In the naive coattention encoder, *max-pooling* was applied on the coattention encodings to obtain the co-dependent representation of the passage,

Proceedings of the 17th International Conference on Natural Language Processing, pages 161–169
Patna, India, December 18 - 21, 2020. ©2020 NLP Association of India (NLPAI)

which forms the base in deciding the relevance. We argue that using max-pooling limits the ability to compose complex co-dependent representations. Intuitively, we can leverage query to supervise the co-dependent representation from coattention encodings of the passage. With this intuition, we propose a simple query-based attention pooling. We show that query-based attention pooling can pick the appropriate clauses which are distributed across the coattention encodings. This allows the model to only focus on relevant parts of passage coattention encodings, which are appropriate in judging the relevance. Additionally, the final passage representation is supervised by the query, which helps the model to better judge the relevance.

To solve these challenges, we first extend the coattention encoder to words and phrases by applying it across all word n-grams of query and passage. For this purpose, similar to C-LSTM (Zhou et al., 2015) and Conv-KNRM (Dai et al., 2018), we generate the word n-gram representations from the word embeddings of query and passage using the convolutional layers of different heights and multiple filters. We then encode these n-gram sequences using a BiLSTM to capture the long term dependencies into n-gram encodings. A coattention encoder is then applied on these n-gram encodings of query and passage to get the coattention encoding of the passage. The coattention encoder first generates the co-dependent representations of query and passage by attending across all word n-grams from query and passage. These co-dependent representations of passage are then fused with the n-gram encodings of the passage using a BiLSTM to get the coattention encoding of the passage. We show that this coattention encoding can better capture the local context between query and passage, thus improving the overall judging power of the model.

To get the final representation of the passage, Alaparthi (2019) applied max-pooling over time on the coattention encoding of the passage (note that coattention encoding is the outputs of BiLSTM). This final representation forms the base in deciding the relevance between query and passage. In this paper, we apply a query based attention pooling on the coattention encoding instead of max-pooling to pick the appropriate clauses which are distributed across the coattention encoding. We argue that, query based attention pooling allows the model to only focus on relevant parts of passage coattention encoding which are appropriate in judging the relevance. Additionally, the final passage representation is supervised by query, which helps the model to better judge the relevance.

We experimented our methods on MS MARCO passage re-ranking task [1] (Bajaj et al., 2016). Making the coattention encoder n-gram aware (uni,bigrams) has increased the Mean Reciprocal Rank @10 (MRR@10) from 28.6 to 29.9 (+4.5% relative increase) when compared to the naive coattention encoder. Replacing the max pooling layer with the query based attention pooling has further improved the MRR@10 to 30.9 (+8.08% overall relative increase), resulting in the best non transformer based model. We show that our methods are competitive to BERT base despite having very less number of parameters. Also, our methods can be easily trained and requires much lesser computational resources.

To summarize, the key contributions of this work are as follows: First, we extend the naive coattention encoder to words and phrases making the coattentions to capture local context. We call it as n-gram coattention encoder. Second, we further extend the n-gram coattention encoder with query based attention pooling to pick the appropriate clauses which are distributed across the coattention encoding of the passage. We call it as n-gram coattention encoder with attention pooling. We show that this can further improve the model in deciding the relevance. Third, we apply our methods on MS MARCO passage re-ranking task and show that our methods have outperformed all the baselines including the previous best non BERT model and are competitive to BERT base. Last, we use examples to compare and discuss our methods with naive coattention encoder.

In section 2, we discuss related work. Then in section 3, we describe our two methods of improving naive coattention encoder. In section 4, we describe the dataset, baselines and the settings we used in all our experiments. Next, we analyze and discuss the results in section 5. Finally, we conclude our work with future plans in section 6.

2 Related Work

Deep learning methods have been successfully applied to a variety of language and information retrieval tasks. By exploiting deep architectures, deep learning techniques are able to discover from training data the hidden structures and features at dif-

[1] https://github.com/microsoft/MSMARCO-Passage-Ranking

ferent levels of abstractions useful for the tasks. Therefore a new direction of Neural IR is proposed to resort to deep learning for tackling the feature engineering problem of learning to rank, by directly using only automatically learned features from raw text of query and passage.

The first successful model of this type is Deep Structured Semantic Model (DSSM) (Huang et al., 2013) introduced in 2013, which is a neural ranking model that directly tackles the adhoc retrieval task. In the same year Lu and Li proposed DeepMatch (Lu and Li, 2013) which is a deep matching method applied to the community question answering and micro-blog matching tasks. Later from 2014 and 2015, there is a rapid increase in neural ranking models, such as new variants of DSSM (Shen et al., 2014), ARC I and ARC II (Hu et al., 2014), MatchPyramid (Pang et al., 2016), etc.,

With the introduction of large scale datasets such as MS MARCO (Bajaj et al., 2016), we have seen a tremendous improvements in neural ranking models. Well known architectures include DUET (Mitra et al., 2017), DUET V2 (Mitra and Craswell, 2019), KNRM (Xiong et al., 2017), Conv-KNRM (Dai et al., 2018), Coattention encoder (Alaparthi, 2019) including the transformer based architectures such as BERT (Nogueira and Cho, 2019), DuoBERT (Nogueira et al., 2019), RepBERT (Zhan et al., 2020).

3 Methodology

In this section, we first describe with notations. Next in section 3.2, we briefly describe the naive coattention encoder first proposed in (Xiong et al., 2016). In section 3.3 and 3.4, we describe our two methods to improve the naive coattention encoder.

Notations Let $Q^{emb} = (x_1^Q, \ldots, x_n^Q) \in \mathbb{R}^{n \times L}$ be the embeddings of words in query of length n, where each word embedding is of dimension L. Similarly, $P^{emb} = (x_1^P, \ldots, x_m^P) \in \mathbb{R}^{m \times L}$ denote the same for words in passage of length m.

3.1 Naive Coattention Encoder

Coattention encoder can be applied on Q^{emb} and P^{emb} to get the coattention encoding of the passage. We first start with encoding the Q^{emb} and P^{emb} using the same BiLSTM (Mueller and Thyagarajan, 2016) to share the representational power:

$$q_t = BiLSTM(q_{t-1}, x_t^Q) \qquad (1)$$

and

$$p_t = BiLSTM(p_{t-1}, x_t^P) \qquad (2)$$

Similar to (Merity et al., 2016; Xiong et al., 2016), we also add sentinel vectors q_ϕ, p_ϕ to allow the query to not attend to any particular word in the passage. So $Q = (q_1, \ldots, q_n, q_\phi) \in \mathbb{R}^{(n+1) \times L}$ and similarly $P = (p_1, \ldots, p_m, p_\phi) \in \mathbb{R}^{(m+1) \times L}$.

Next, we compute the affinity scores between all pairs of query and passage words: $L = P^T Q$. We call L as affinity matrix. The affinity matrix is normalized row wise to get the attention weights A^Q across the passage for each word in query. Similarly, normalized column wise to get the attention weights A^P across the query for each word in the passage:

$$A^Q = softmax(L) \qquad (3)$$

and

$$A^P = softmax(L^T) \qquad (4)$$

Next, we compute the attention contexts, of the passage in light of each word in the query:

$$C^Q = PA^Q \qquad (5)$$

Additionally, we compute the summaries $C^Q A^P$ of the previous attention contexts in light of each word in the passage. We also define C^P, a co-dependent representation of the query and passage, as the coattention context:

$$C^P = [Q; C^Q] A^P \qquad (6)$$

Here $[x; y]$ is concatenation of vectors x and y horizontally. The last step is the fusion of temporal information to the coattention context via a bidirectional LSTM:

$$u_i = BiLSTM_{fusion}(u_{i-1}, u_{i+1}, [p_i; c_i^P]) \qquad (7)$$

We define $U = [u_1, \ldots, u_m]$, the outputs of $BiLSTM_{fusion}$ concatenated vertically, as the coattention encoding of the passage. Here $U \in \mathbb{R}^{m \times L'}$ and L' is the dimension of the hidden state in $BiLSTM_{fusion}$.

For the rest of this paper, we treat the coattention encoder as a module, defined as:

$$U = CoAttentionEncoder(Q^{emb}, P^{emb}) \qquad (8)$$

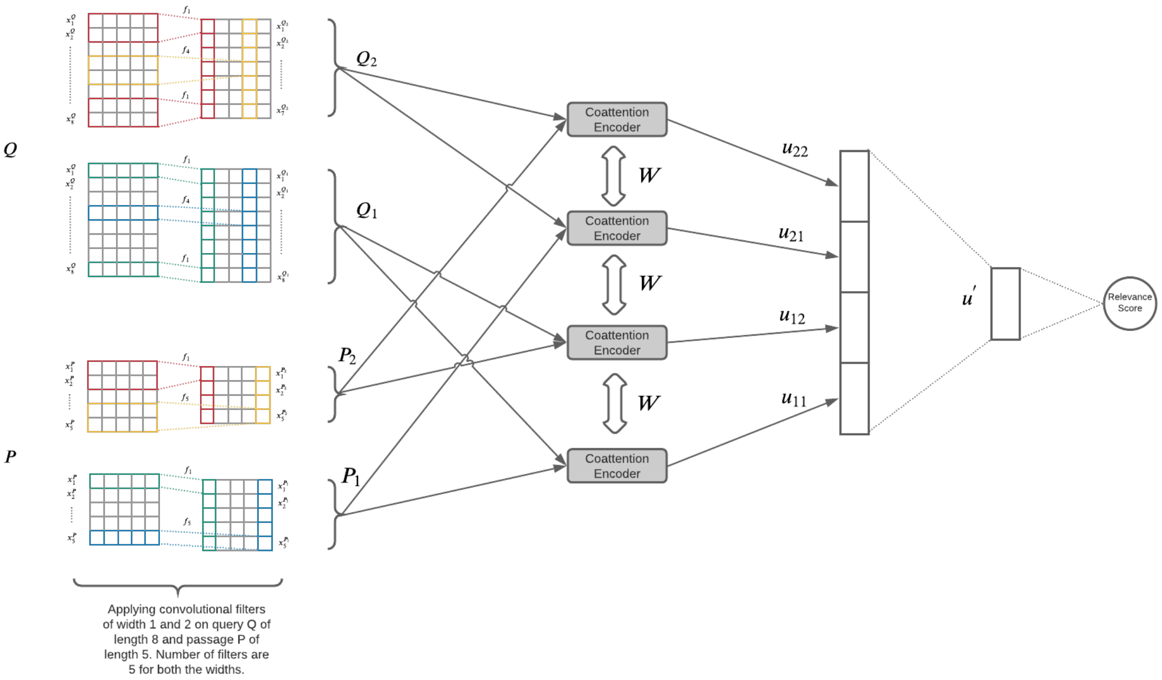

Figure 1: Architecture of ngram aware coattention encoder

3.2 Word N-Gram Coattention Encoder

In this section, we extend the naive coattention encoder by applying it across all word n-grams from query and passage as shown in Figure 1. To compute the word h-gram representations of query Q^{emb}, where each h-gram representation is of dimension F, similar to (Zhou et al., 2015; Dai et al., 2018), we apply F convolution filters of height h and width L. Note that L is the dimension of the word embeddings. For each window of h words, a single filter $filter_f$ performs a weighted sum of all word embeddings $x^Q_{t:t+h}$ parameterized by it's weights $w_f \in \mathbb{R}^{hL}$ and bias $b_f \in \mathbb{R}$:

$$v^h_f = w_f \cdot x^Q_{t:t+h} + b_f, v_f \in \mathbb{R} \qquad (9)$$

Using F filters, we get F scores $v^h_1, .., v^h_F$, each describing $x^Q_{t:t+h}$ in a different perspective. These v^h_f from F filters are concatenated into a single vector and $tanh$ activation is then applied to get the F-dimensional embedding:

$$x^{Q_h}_t = tanh([v^h_1; \ldots; v^h_F]) \in \mathbb{R}^F, t = 1..n{-}h{+}1 \qquad (10)$$

We define h-gram sequence of the query as

$$Q_h = [x^{Q_h}_1, ..., x^{Q_h}_{n-h+1}] \qquad (11)$$

Note that padding is not applied to the sequence. Similarly, we apply the same convolution filters to get the h-gram representations of passage P^{emb}:

$$P_h = [x^{P_h}_1, \ldots, x^{P_h}_{m-h+1}] \qquad (12)$$

Here $Q_h \in \mathbb{R}^{(n-h+1)\times F}$ and $P_h \in \mathbb{R}^{(m-h+1)\times F}$. Using these convolutional layers of different heights, we get different n-gram sequences.

Coattention encoder is applied on all Q_i and P_j $\forall i, j \in [1..H]$, H is a parameter, which denotes the span of the n-gram. In this paper, we have only experimented with uni,bi-grams i.e., $H = 2$. Note that, $H = 1$ reduces to naive coattention encoder i.e., unigrams/words. Coattention encoder applied on Q^i and P^j generates a coattention encoding of the passage denoted by:

$$U_{ij} = CoAttentionEncoder(Q_i, P_j) \qquad \forall i, j \in [1..H] \qquad (13)$$

Here $U_{ij} \in \mathbb{R}^{(m-j+1)\times L'}$ and to recall,

$$U_{ij} = [u_{ij_1}, u_{ij_2}, \ldots, u_{ij_{m-j+1}}] \qquad (14)$$

where u_{ij_t} is the output from the $BiLSTM_{fusion}$ at time step t. We call U_{ij} as the coattention encoding of the passage P_j with respect to the query Q_i.

To get the relevance score, similar to (Alaparthi, 2019), a max-pooling layer over time can be applied on U_{ij} to get the single representation (single thought vector):

$$u_{ij} = max(\{u_{ij_t}\}_{t=1..m-j+1}) \in \mathbb{R}^{L^{'}} \qquad (15)$$

We concatenate these representations $u_{ij} \forall i, j \in [1..H]$ horizontally to get a single vector $u^{'}$ a n-gram aware coattention representation of passage.

$$u^{'} = [u_{11}; u_{12}; \ldots ; u_{H-1H}; u_{HH}] \in \mathbb{R}^{H^2 L^{'}} \qquad (16)$$

The $u^{'}$ is then passed to a linear layer parameterized by weights $W_s \in \mathbb{R}^{H^2 L^{'}}$ to get the relevance score:

$$score_{P|Q} = W_s^T u^{'} \qquad (17)$$

3.3 Coattention Encoder with Attention Pooling

In the naive coattention encoder and in the previous section, max-pooling over time is applied on the coattention encoding to get the single representation capturing entire sense of the passage. In this section, we propose a simple attention pooling to select key parts from the coattention encoding U_{ij} of the passage using the query: $q^{'} = q_{i_{n^{'}}}$, $n^{'} = n - i + 1$. We also add sentinel vector $d^{'}_{\phi}$ (Merity et al., 2016) to U_{ij} to allow the query to not attend to any particular clause in the passage:

$$u_{ij} = \sum_{t=1}^{t=m-j+1} \alpha_t u_{ij_t} \qquad (18)$$

Where,

$$\alpha_t = \frac{exp(u_{ij_t}^T q^{'})}{\sum_{k=1}^{k=m-j+1} exp(u_{ij_k}^T q^{'})} \qquad (19)$$

Similar to previous section, $u_{ij} \forall i, j \in [1..H]$ can then be concatenated horizontally into a single vector $u^{'}$ and this $u^{'}$ can be used to get the relevance score.

4 Experimental Setup

This section describes our datasets, how training and testing were performed and our implementation details.

4.1 Dataset and Learning rule

We perform all our experiments on Microsoft MAchine Reading COmprehension (MS MARCO) passage re-ranking task. The whole corpus consists of 8.8M passages extracted from 3.6M web documents corresponding to 500K anonymized user queries sampled from Bing's search query logs.

For the ease of training, MS MARCO team has released a pre-processed training set *triples.train.small.tsv*[1], which contain the triples $\langle Q, P^+, P^- \rangle$, where Q is the query, P^+ and P^- are passages, P^+ being more relevant. We train all our models on *triples.train.small.tsv*[2]. We use the Cross Entropy loss employed by a softmax function on relevance scores $score_{P+|Q}$ and $score_{P-|Q}$ to learn the parameters Θ of the models. Some subset of query, passages are randomly chosen from *top1000.dev.tsv*[2] to tune the models. Finally, we predict the ranks on *top1000.eval.tsv*[2].

$$L(\Theta) = - \sum_{\langle Q,P^+,P^- \rangle \in \mathbf{S}} logP(P^+ | \langle Q, P^+, P^- \rangle)) \qquad (20)$$

where,

$$P(P^+ | \langle Q, P^+, P^- \rangle)) = \frac{exp(score_{P+|Q})}{exp(score_{P+|Q}) + exp(score_{P-|Q})} \qquad (21)$$

4.2 Hyperparameters

In all our experiments, we use FastText (Bojanowski et al., 2017) word embeddings of dimension 300. These FastText embeddings are trained on all queries and passages from the training set, we freeze these embeddings during the training. All the other parameters of the model are initialized using an uniform distribution $U(-0.01, 0.01)$. The number of filters in convolution layers is set to 300. We only experiment with uni and bi-grams i.e, $H = 2$. We use the BiLSTMs with 2 layers and hidden sizes of 512 with dropout of 0.2 (Srivastava et al., 2014) between the layers. ADAM optimizer (Kingma and Ba, 2014) with initial learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ is used. We truncate the query and passage lengths to 30 and 150 words, train our network until convergence with batch size of 128. On a single 1080 Ti machine, training takes around 8 hours to converge. We evaluate our model on dev set every 500 steps and decay the learning rate by a factor of 0.5 every 5,000 steps.

[2] https://github.com/microsoft/MSMARCO-Passage-Ranking#data-information-and-formating

Method	**MRR@10 Dev**	**MRR@10 Eval**	**Parameters**
KNRM (Xiong et al., 2017)	21.8	19.8	-
Duet V2 (Mitra and Craswell, 2019)*	24.3	24.5	-
Conv-KNRM (Dai et al., 2018)	24.7	24.7	-
FastText + Conv-KNRM (Hofstätter et al., 2019)	29.0	27.1	-
IRNet **	27.8	28.1	-
Naive coattention encoder (Alaparthi, 2019)	28.8	28.6	6.9M§
N-gram coattention encoder **(Ours)**	31.0	29.9 **(+4.54%)**	9.6M §
+ attention pooling **(Ours)**	31.9	30.9 **(+8.08%)**	9.6M §
BERT Base	34.4	33.5	109M
BERT Large (Nogueira and Cho, 2019)	36.5	35.8	340M

Table 1: Comparison of the different methods. The variants of the coattention encoder benefits significantly from the modifications described in this paper. * Official Baseline; ** Unpublished work; § These do not include the parameters from word embeddings as we directly use the pre-trained FastText embeddings and do not update them during the training.

5 Results and Discussion

In this section, we present the evaluation results of our models and compare our models with various baselines.

5.1 Comparision with Baselines

Table 1 lists the MRR@10 scores on Dev and Eval sets. We compare the naive coattention encoder and two proposed methods with the baselines including BERT. From the table, we get the following observations: (1) Firstly, the naive coattention encoder has performed better than the existing best non BERT based models: Conv-KNRM and IRNet (2) Applying the coattention encoder on uni-grams and bi-grams resulted in an increase in MRR@10 on eval from 28.6 to 29.9 (relative increase of 4.5%). This indicates that the model can capture the more robust interactions between query and passage. (3) Using the query based attention pooling instead of max-pooling over time further increased the score by 3.3% indicating that the model can now focus on the appropriate clauses in the passage leading to better passage representation and appropriate relevance score. (4) We can also observe that despite having less number of parameters compared to BERT, (9.6M vs. 109M), our models are competitive to BERT (30.9 vs. 33.5).

5.2 Analysis with respect to query type

Type	**# Queries**	**A**	**B**	**C**
what	2751	28.07	29.25	**30.85**
others	2274	31.21	32.17	**33.86**
how	837	23.28	23.8	**25.73**
where	283	37.6	37.69	**39.32**
who	278	28.9	29.91	**33.66**
when	189	23.42	**26.38**	23.92
define	173	27.84	**29.21**	28.31
which	120	21.03	22.35	**23.53**
why	75	19.55	**23.31**	22.92

Table 2: Comparision of naive coattention encoder (**A**) with the two variants described in this paper with respect to query type. In the table, **B** represents the n-gram coattention encoder and **C** represents the n-gram coattention encoder with attention pooling.

Table 2 lists the MRR@10 scores of naive coattention encoder (represented by **A**), n-gram coattention encoder (represented by **B**) and n-gram coattention encoder with attention pooling (represented by **C**) which were evaluated using Dev set.

From the table, we can observe that, both the n-gram coattention encoders consistently outperformed the naive coattention encoder. It is interesting to see that plain n-gram coattention encoder (with out attention pooling, represented by column **B**) outperformed the n-gram coattention encoder with attention pooling (column **C**) in case of *when*, *define*, *why* type queries.

5.3 Qualitative Analysis

Table 3 lists the best passages ranked by the naive coattention encoder and our methods described in this paper for the various queries. In this section, we qualitatively analyze the performances of the coattention encoders.

Query	Method	Best Ranked passage
who is tom cavanagh?	Naive coattention encoder	For other people named Tom Corbett, see Tom Corbett (disambiguation). Thomas Wingett Tom Corbett, Jr. (born June 17, 1949) is an American politician and attorney who served as the 46th Governor of Pennsylvania from January 18, 2011 to January 20, 2015. He is a member of the Republican Party.
	N-gram coattention encoder	Tom Cavanagh on Why Grant Gustin Deserves to Be THE FLASH in the Movies, Too. Share: Tom Cavanagh is a national treasure. No, not just because he is the Tom in our beloved Mike and Tom Eat Snacks podcast, or because of his chilling performance as Dr. Harrison Wells on The CW's The Flash. But rather because the Canadian actor is unafraid to speak his mind — which often happens to coincide with exactly what we were thinking, too.
	N-gram coattention encoder with query attention	Grodd (via Harrison Wells) Thomas Patrick Tom Cavanagh (born October 26, 1963) is a Canadian actor. He portrays the various iterations of Harrison Wells on The Flash.
what is January birthstone color	Naive coattention encoder	November Birthstone Color. The November birthstone color is usually light to dark yellow, however, topaz, the official November birthstone comes in a range of great colors such as several shades of yellow, pale green, blue, red, pink, black, and brown. Pure topaz is actually a colorless stone. The red and pink topaz gets their color from chromium.
	N-gram coattention encoder	Birthstone color list. January Birthstone Color. The birthstone for the month of January is the garnet, which means that red is the commonly accepted January birthstone color. It signifies trust and friendship, which makes it a good gift for a friend. The word garnet comes from the Latin word granatum, which means pomegranate.
	N-gram coattention encoder with query attention	Birthstone color list. January Birthstone Color. The birthstone for the month of January is the garnet, which means that red is the commonly accepted January birthstone color. It signifies trust and friendship, which makes it a good gift for a friend. The word garnet comes from the Latin word granatum, which means pomegranate.
why was napalm used in the vietnam war	Naive coattention encoder	Napalm. Napalm is jellied gasoline. Its name is an acronym of naphthenic and palmitic acids, which are used in its manufacture. While used in World War II and the Korean War, napalm became notorious in Vietnam where it was used in three capacities.Possibly its most visual use was being dropped from aircraft in large canisters which tumbled sluggishly to earth.apalm is jellied gasoline. Its name is an acronym of naphthenic and palmitic acids, which are used in its manufacture.
	N-gram coattention encoder	Napalm. U.S. troops used a substance known as napalm from about 1965 to 1972 in the Vietnam War; napalm is a mixture of plastic polystyrene, hydrocarbon benzene, and gasoline. This mixture creates a jelly-like substance that, when ignited, sticks to practically anything and burns up to ten minutes.
	N-gram coattention encoder with query attention	The US first used napalm during World War II in both the European and Pacific theaters, and also deployed it during the Korean War. However, those instances are dwarfed by American use of napalm in the Vietnam War, where the US dropped almost 400,000 tons of napalm bombs in the decade between 1963 and 1973. Of the Vietnamese people who were on the receiving end, 60% suffered fifth degree burns, meaning that the burn went down to the bone.
what energy source is earth using primarily for its internal heat	Naive coattention encoder	5. According to the lecture, what energy source is Earth using primarily for its internal processes? a. [Interior heat] b. [Geothermal energy] c. [Solar energy] d. [Radioactive Decay] e. [Magma] 6. According to the lecture, what energy source is Earth using primarily for its external/surficial processes? a. [Interior heat] b. [Geothermal energy] c. [Solar energy] d.
	N-gram coattention encoder	5. According to the lecture, what energy source is Earth using primarily for its internal processes? a. [Interior heat] b. [Geothermal energy] c. [Solar energy] d. [Radioactive Decay] e. [Magma] 6. According to the lecture, what energy source is Earth using primarily for its external/surficial processes? a. [Interior heat] b. [Geothermal energy] c. [Solar energy] d.
	N-gram coattention encoder with query attention	5. According to the lecture, what energy source is Earth using primarily for its internal processes? a. [Interior heat] b. [Geothermal energy] c. [Solar energy] d. [Radioactive Decay] e. [Magma] 6. According to the lecture, what energy source is Earth using primarily for its external/surficial processes? a. [Interior heat] b. [Geothermal energy] c. [Solar energy] d.

Table 3: Best ranked passages by naive coattention encoder and it's 2 variants

Considering the query *Who is tom cavanagh?*, we can notice that naive coattention encoder although ranked the passage which semantically answers the query, it utterly fails as Tom Corbett and Tom Cavanagh are completely different persons. Although n-gram aware coattention encoder was able to mark the passage containing Tom Cavanagh as relevant, but it could not correctly capture the required sense from the passage. Finally, adding the query attention to the n-gram coattention encoder improved the ranking performance as we can see that the model was now able to correctly rank the passage.

Similarly, in case of query *what is January birthstone color*, naive coattention encoder has marked the passage relevant which corresponds to November birthstone color. However, both the n-gram coattention encoders have marked the correct passage as relevant, which is related to *January birthstone*. These two examples suggests that n-gram coattention encoders are able to correctly capture the local context and can capture the robust interactions between query and passages, thus improving the overall model performance.

In case of third query *why was napalm used in the vietnam war*, the naive coattention encoder predicted the passage containing the terms napalm, vietnam war. But the passage does not answer the query. The n-gram coattention with attention pooling marks the passage as relevant which describes about the effects napalm has created in vietnam war but the passage does not correctly answer the reason for using napalm in vietnam war. Interestingly, the n-gram coattention with out attention pooling predicted the correct relevant passage.

Lastly, for the query *what energy source is earth using primarily for its internal heat*, all the coattention encoders predicted a passage which is not relevant. One interesting observation is that, the query is part of the passage itself. This shows that the coattention mechanism has trouble discriminating the passage which is semantically similar to the query but does not have an answer in it.

6 Conclusion and Future Work

In this paper, we proposed two simple extensions to naive coattention encoder, namely, n-gram coattention encoder which attends the words and word n-grams to better capture the interactions between query and passage. Later, we proposed simple attention pooling to pick the appropriate clauses which are distributed across the coattention encoding of the passage. Our experiments on MS MARCO passage re-ranking task shows that our models outperformed all the baselines including the naive coattention encoder. We also compare our methods with BERT and show that our methods are competitive to BERT base despite having very less number of parameters, thus our models are very easy to train and are computationally efficient.

We have also compared the performance of coattention encoders with respect to the query types and also qualitatively analyzed the performances by taking few examples. We show that n-gram coattention encoders now capture the local context very well and also show the delimitation of coattention mechanism.

In the future, we would like to perform more deeper analysis on delimitations of coattention mechanism. Apart from this, our future line of research would be as follows: Incorporating the handcrafted features such as BM25 and study the performance. It would be interesting to see how the performance of the models will change with respect to the context based embeddings such as ELMo, BERT, etc.,

References

Chaitanya Sai Alaparthi. 2019. Microsoft ai challenge india 2018: Learning to rank passages for web question answering with deep attention networks. *arXiv preprint arXiv:1906.06056*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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*, pages 126–134.

Sebastian Hofstätter, Navid Rekabsaz, Carsten Eickhoff, and Allan Hanbury. 2019. On the effect of low-frequency terms on neural-ir models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1137–1140.

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*, pages 2042–2050.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

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*, pages 279–287.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pages 289–297.

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Advances in neural information processing systems*, pages 1367–1375.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Bhaskar Mitra and Nick Craswell. 2019. An updated duet model for passage re-ranking. *arXiv preprint arXiv:1903.07666*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *thirtieth AAAI conference on artificial intelligence*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. *arXiv preprint arXiv:1602.06359*.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web*, pages 373–374.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406.

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*, pages 55–64.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.