Hermitian Co-Attention Networks for Text Matching in Asymmetrical Domains

Yi Tay[1], **Anh Tuan Luu**[2], **Siu Cheung Hui**[3]
[1,3] Nanyang Technological University, Singapore
[2] Institute for Infocomm Research, A*Star, Singapore
ytay017@e.ntu.edu.sg, at.luu@i2r.a-star.edu.sg, asschui@ntu.edu.sg

Abstract

Co-Attentions are highly effective attention mechanisms for text matching applications. Co-Attention enables the learning of pairwise attentions, i.e., learning to attend based on computing word-level affinity scores between two documents. However, text matching problems can exist in either symmetrical or asymmetrical domains. For example, paraphrase identification is a symmetrical task while question-answer matching and entailment classification are considered asymmetrical domains. In this paper, we argue that Co-Attention models in asymmetrical domains require different treatment as opposed to symmetrical domains, i.e., a concept of word-level directionality should be incorporated while learning word-level similarity scores. Hence, the standard inner product in real space commonly adopted in co-attention is not suitable. This paper leverages attractive properties of the complex vector space and proposes a co-attention mechanism based on the complex-valued inner product (Hermitian products). Unlike the real dot product, the dot product in complex space is asymmetric because the first item is conjugated. Aside from modeling and encoding directionality, our proposed approach also enhances the representation learning process. Extensive experiments on five text matching benchmark datasets demonstrate the effectiveness of our approach.

1 Introduction

Computing relevance scores between textual documents (a.k.a text matching) is a widely researched area in natural language processing and information retrieval. The wide interest in this topic is understandable, given that text matching enables a broad spectrum of applications ranging from question-answer retrieval systems to paraphrase identification. The real world applications of text matching is also broad, encompassing possibilities such as automated FAQ systems or microblog retrieval. Our work is concerned with the general application of text matching, focusing on short sentences as documents.

Recent advances and state-of-the-art in this field comprise mainly neural models. Neural networks (or deep learning) are trained end-to-end to predict the relevance between documents which may be used to serve ranked lists during inference. In most architectures, networks are *siamese* in nature, using identical encoders for both documents. In relatively simpler models, encoded representations are typically combined using a parameterized function such as a feed-forward neural network or non-parameterized functions such as cosine similarity. This may be extended, in which representations are learned pairwise using recent advances such as grid-wise feature aggregation and co-attentional mechanisms.

This paper is based on an observation that not all text matching tasks are created equal. We characteristically dichotomize them into two categories - symmetrical problems and asymmetrical problems. Examples of symmetrical problems include paraphrase identification and semantic textual similarity in which the positions of documents do not matter, i.e., $s(a, b) = s(b, a)$. Asymmetrical domains, conversely, include problems such as answer retrieval and entailment classification in which $s(a, b) \neq s(b, a)$. Consider the following (negative) example in question-answer retrieval.

1. *'May I know where IJCAI 2018 will be located?'*
2. *'It is where top AI researchers present their latest works.'*

Clearly, we notice that the word *'where'* has vastly different semantics just based on whether it appears in the question (1) or answer (2). Hence, a sense of directionality is critical, i.e., allowing models to have a sense of $a \rightarrow b$. This applies to many other words in which the relative importance largely depends on whether it belongs to a or b. It is also intuitive that directionality is important due to the lexical gap and a common need for co-reference resolution in QA pairs. While a straightforward and naive way is to decouple parameters between a and b, this has generally been found to not only incur extra parameter costs but also degrade performance. As such, many text matching methods do not distinguish between a and b and compute relevance scores indiscriminately.

This paper proposes a novel co-attentional mechanism with two key benefits - (1) inducing a sense of directionality and (2) enhancing representation learning. We investigate complex-valued inner product to model word-word similarities when learning co-attentions. This exploits the property of complex vector space, i.e., $\mathbb{C}$ where the inner product (Hermitian inner product) is actually asymmetrical because the first matching sentence is conjugated. Moreover, our new co-attention mechanism also benefits from expanded representation capability, an inherent advantage brought by complex vector spaces.

Typically, Co-Attention models similarity between a and b indiscriminately, i.e., $s(a, b) = s(b, a)$. In other words, all word pairs return the same scores irregardless of position. We further elaborate on the potential weaknesses of this. Consider the following entailment pair from the SciTail [Khot *et al.*, 2018] dataset:

1. *'A concave lens is thinner in the middle than it is near its edges'*
2. *'A concave lens is thicker at the edges than it is in the middle.'*

In the above example, complex reasoning is required to successfully classify this pair. However, there are many repeated words, e.g., *middle*, *edges*, etc. Intuitively, this raises the need for the model to *maintain* some forms of positional information when matching each individual word pairs. As such, it is helpful that a model is able to differentiate whether the words - *thicker*, *thinner*, *middle*, *edges* come from a or b.

1.1 Our Contributions

The overall contributions of this paper are summarized as follows:

- We propose a novel Hermitian Co-Attention (HCA) mechanism for asymmetrical text matching problems. We propose an overall model architecture, the Hermitian Co-Attention Recurrent Network (HCRN) for text matching. We demonstrate the utility of complex-valued co-attention mechanisms.
- We conduct extensive experiments on five benchmark datasets in four domains of entailment classification (SciTail), question answer retrieval (TrecQA, WikiQA), Twitter Customer Support and Dialogue Prediction (Ubuntu Corpus). HCRN achieves highly competitive results on all datasets. HCRN outperforms state-of-the-art models such as BiMPM [Wang *et al.*, 2017], ESIM[Chen *et al.*, 2017] and KEHNN [Wu *et al.*, 2016] on their respective tasks. The results reflect a considerable gain over when only vanilla co-attention mechanism is used.

2 Related Work

Text matching is a core research problem in NLP and Information Retrieval. Many problems that require a relevance score to be computed between two documents (or sentences) can be cast as a text matching problem. A wide range of problems fall into this problem formulation such as question answering [Yang *et al.*, 2015], document search [Shen *et al.*, 2014], entailment classification [Khot *et al.*, 2018] ,paraphrase identification [Wang *et al.*, 2017] and recommendation with reviews [Tay *et al.*, 2018b]. As such, general purpose text matching algorithms are highly attractive as they can be applied to a diverse range of applications.

The dominant state-of-the-art methods today are mostly based on deep learning. Early work in this paradigm explores recurrent [Wu *et al.*, 2016; Wang *et al.*, 2016a] and convolutional based encoders [Hu *et al.*, 2014; Severyn and Moschitti, 2015] for document representation and subsequently learns a similarity function between these representations. Several works explore alternate encoders such as recursive networks [Wan *et al.*, 2016b] and quasi-recurrent networks [Tay *et al.*, 2017c]. A wide range of parameterized similarity functions have been explored such as multi-layered perceptrons [Severyn and Moschitti, 2015], neural tensor networks [Qiu and Huang, 2015] and holographic hidden layers [Tay *et al.*, 2017a]. Recent advances in question answer matching have considered several novel matching functions such as Hyperbolic distance [Tay *et al.*, 2018a] and quantum-like language models [Zhang *et al.*, 2018].

Recent advances exploit two main paradigms. The first paradigm utilizes extensive matching operations [Wan *et al.*, 2016a; He *et al.*, 2015; Wang *et al.*, 2017] and aggregates an overall matching vector(s) for prediction. The Bilateral Multi-Perspective Matching (BiMPM) is one of the recent state-of-the-art models for general text matching, utilizing a multi-perspective cosine matching function to model across multiple views. The second paradigm mainly utilizes co-attention [Xiong *et al.*, 2016] to learn pairwise attentions. A diverse range of co-attention mechanisms exist, mainly varying the innovation at the similarity matrix computation layer and pooling of this similarity matrix. Attentive pooling [Santos *et al.*, 2016] is a form of *extractive* co-attention, using the *max* pooling operator to extract strong signals in the documents. Models that utilize alignment-based pooling [Parikh *et al.*, 2016; Wang and Jiang, 2016] are also prominent especially in the areas of entailment classification. Unlike the max pool operator, alignment-based pooling learns subphrase alignments between two documents. The ESIM (Enhanced Sequential Inference Model) [Chen *et al.*, 2017], which utilizes alignment-pooling is a strongly competitive model for entailment classification. Recently, [Tay *et al.*, 2017b] proposed CAFE, a new alignment-pooling model with factorization layers and achieved state-of-the-art performance on entailment classification.

The innovation of this work lies in the computation of affinity scores within Co-Attention mechanisms. [Parikh *et al.*, 2016] adopted feed-forward neural networks to first transform words and then used the inner product to compute scores between word-pairs. The ESIM [Chen *et al.*, 2017] used the plain inner product in lieu of the fact that the words have been encoded in previous layers by a recurrent model. Attentive pooling [Santos *et al.*, 2016] used a bilinear scoring function to compute the affinity matrix. This work investigates the usage of scoring functions that are both asymmetrical and expressive, exploiting computation in Complex space $\mathbb{C}$ for computing this affinity matrix.

Many works have demonstrated the utility of complex-valued parameters. Our work is inspired by ComplEx [Trouillon *et al.*, 2016], a knowledge base embedding method that uses the complex inner product to explicitly model asymmetric relations in knowledge bases. Notably, ComplEx achieved state-of-the-art performance, by simply replacing real-valued dot products with complex parameters. [Danihelka *et al.*, 2016] proposed complex-valued long short-term memory networks, drawing links to holographic reduced representations [Plate, 1995] and associative memory. Complex-valued parameters have been used to parameterize recurrent models [Arjovsky *et al.*, 2016] as a novel strategy to combat vanishing gradients and also enhance its representation capacity. A recent work [Trabelsi *et al.*, 2017] proposed various building blocks for complex networks such as activation functions and convolutions. Contrary to these works, our work aims to demonstrate the effectiveness of isolated complex modules within real-valued neural networks, i.e., in our proposed approach, only the co-attention mechanism operates in complex vector space. This saves parameter cost by only using complex-valued matching where it is most necessary.

3 Our Proposed Approach

In this section, we describe our proposed model. Figure 1 describes our model architecture.

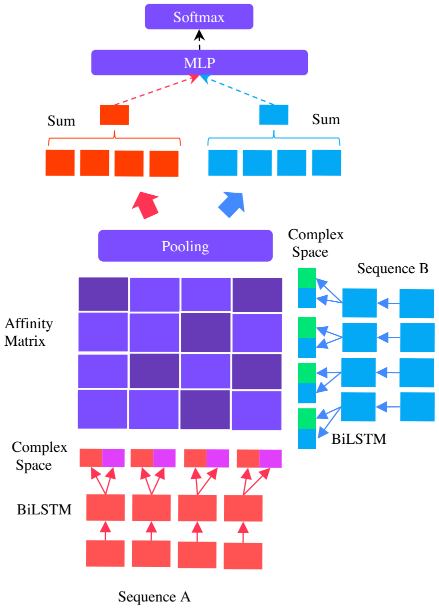

Figure 1: Our Proposed Model Architecture with Softmax Activation.

3.1 Input Encoding

Our network accepts two sequences (a, b) as an input. For the sake of brevity, we only describe the encoding of one sequence since the input encoder is functionally symmetrical. Each input is represented as a sequence of one hot encoded vectors, $\{w_1, \cdots w_\ell\} \in \mathbb{R}^{|V|}$. An embedding matrix $\mathbf{W}_e \in \mathbb{R}^{r \times |V|}$ converts each word into a r-dimensional vector. Each word is passed into a projection layer with ReLU activation as follows:

$$u_i = ReLU(W_p w_i) + b_i \tag{1}$$

where $W_p \in \mathbb{R}^{r \times n}, b_i \in \mathbb{R}^n$ are the parameters of the projection layer. Next, each sequence is then passed through a bidirectional long short-term memory (LSTM) encoder.

$$h_i = \text{BiLSTM}(u, i), \forall i \in [1, \dots \ell] \tag{2}$$

where ℓ represents the maximum length of the sequence. The bidirectional LSTM runs a single-directional LSTM encoder in two directions (forward and backward). The output at each timestep h_i, is the concatenation of the hidden states from both directions. Notably, the parameters of the BiLSTM are *siamese* in nature, sharing weights between all input sequences. The output of the input encoding layer is therefore, $\{h_1, \cdots h_\ell\} \in \mathbb{R}^d$ where $d = 2n$.

3.2 Vanilla Co-Attention

First, we introduce the baseline symmetrical (Vanilla Co-Attention) mechanism. The inputs to this layer are the encoded representations from the BiLSTM layer. For simplicity, we refer to these encoded representations as $\bar{a}, \bar{b}$. There are many variants of Co-Attention mechanisms. As an example, we consider the alignment adaptation proposed by [Parikh *et al.*, 2016]. Let $\bar{a}$ and $\bar{b}$ be sequence pairs. A similarity (affinity) matrix $s \in \mathbb{R}^{\ell_a \times \ell_b}$ is formed by passing each word through $F(.)$ as follows:

$$s_{ij} = F(\bar{a}_i)^\top \cdot F(\bar{b}_j) \tag{3}$$

where $s \in \mathbb{R}^{\ell_a \times \ell_b}$ and $\bar{a}_i, \bar{b}_j$ are the i-th and j-th word in the a and b respectively. $F(.)$ is a single-layered feed-forward neural network $f(x) = ReLU(W(x) + b)$. There are various pooling layers that could be utilized such as alignment pooling or extractive pooling which are described as follows:

1. **Alignment Pooling** learns to align sub-phrases of two sequences together. The alignment pooling operation is defined as:

$$\beta_i = \sum_{j=1}^{\ell_a} \frac{exp(s_{ij})}{\sum_{k=1}^{\ell_a} exp(s_{ik})} \bar{a}_j \;\; ; \;\; \alpha_j = \sum_{i=1}^{\ell_b} \frac{exp(s_{ij})}{\sum_{k=1}^{\ell_b} exp(s_{kj})} \bar{b}_i \tag{4}$$

where β_i is the sub-phrase in $\bar{a}$ that is softly aligned to a_j. Intuitively, β_i is a weighted sum across $\{a_j\}_{j=1}^{\ell_a}$, selecting the most relevant parts of $\bar{a}$ to represent β_i.

2. **Extractive Pooling** Alternatively, an extractive pooling may also be performed by taking column-wise and row-wise *max* pooling across s.

$$a' = S(\max_{col}(s))^\top a \quad \text{and} \quad b' = S(\max_{row}(s))^\top b \tag{5}$$

where $s \in \mathbb{R}^{\ell_a \times \ell_b}$ is the affinity matrix. $S(.)$ is the softmax function. a', b' are the co-attentional representations of a and b respectively. Intuitively, max pooling selects each word based on its maximum importance of all word in the other text.

The choice of pooling operator is task-dependent and is tuned as a hyperparameter, in similar spirit to how pooling operators are tuned in LSTM or CNN models.

3.3 Hermitian Co-Attention

In this section, we introduce our novel co-attention mechanism. This variation is inspired by complex-valued representations. Interestingly, the complex-valued dot product (also known as the Hermitian Inner Product or *sesquilinear form*) is non-commutative which is defined as:

$$\langle a_i, b_j \rangle = \bar{a}_i{}^\top b_j \tag{6}$$

where a_i, b_j are complex-valued vectors, i.e., $a = Re(a_i) + iIm(a_i)$ where $Re(a_i)$ and $Im(a_i)$ are the real and imaginary parts of the vector. i is the square root of -1. $\bar{a}_i$ is the complex conjugate of the complex vector a_i. Note that the complex conjugate over the first vector in the Hermitian product makes it asymmetric, i.e., $\langle a, b \rangle \neq \langle b, a \rangle$, which is a property that we are seeking in our novel Co-Attention model.

Complexification

Our network introduces a brief complexification process in the network. In other words, while the Hermitian Co-Attention module operates in complex vector space, the inputs and outputs are real-valued vectors. This is in similar spirit to [Trouillon *et al.*, 2016] that extracts the real component for prediction. In order to initialize the imaginary component, we use a nonlinear transform layer to project[1] the initial inputs to another vector space. Subsequently, we complexify both real vector spaces into a complex vector space $\mathbb{C}$.

Similarity Matrix Computation

Finally, similarity matrix is computed by:

$$s_{i,j} = Re(\langle a_i + iF_{proj}(a_i), b_j + iF_{proj}(b_j) \rangle) \tag{7}$$

where $\langle ., . \rangle$ is the Hermitian inner product and $Re(.)$ denotes the real component of the complex-valued matrix. Alternatively, we also explore the complex bilinear product.

$$s_{i,j} = Re(a_i^\top \mathbf{M}\, b_j) \tag{8}$$

where $a_i, b_j \in \mathbb{C}^d$ and $M \in \mathbb{C}^{d \times d}$. The existence of this matching matrix $\mathbf{M}$ is tuned as a hyperparameter in our experiments.

Isolated Complex Module

Different from entirely complex-valued neural networks, our network only exploits a partially complex module. There are two good reasons for this. Firstly, we want to avoid the mandatory incorporation of complex-differentiable and holomorphic activation functions which are required to handle complex-valued input-outputs. In this case, complex-specific versions of activation functions such as ReLU have to be used [Trabelsi *et al.*, 2017]. In our case, the complex module is self-contained and converted to real vectors before any activation function is applied. Secondly, we want to minimize the parameter cost incurred by using entirely complex-valued networks. Hence, our network is only complex-valued within the co-attention module where properties of complex spaces are most desired.

3.4 Hermitian Intra-Attention (Optional)

Following [Parikh *et al.*, 2016], intra-attention (or self-attention), when applied individually to each sentence, can help improve awareness of each sentence to the entirety of its context. The Intra-Attention function is defined as:

$$x'_i = \sum_{j=1}^{\ell} \frac{exp(\hat{s}_{ij})}{\sum_{k=1}^{\ell} exp(\hat{s}_{ik})} x_j \tag{9}$$

where x'_i is the intra-attentional representation of x_i. Note that following the Hermitian Co-Attention, we also use the complex-valued inner product to calculate $\hat{s}_{ij}$. The output of the intra-aligned output x'_i is concatenated to the original x_i, i.e., $[x'_i; x_i]$. The intra-attention layer is applied right after the input encoding layer. This layer is optional and we found varying results in performance when applying this layer. As such, this is also tuned.

3.5 Aggregation and Prediction

After the co-attention layer, we sum (aggregate) the weighted representations to form two d-dimensional representations. The prediction layer is dependent on the dataset and is described as follows:

- **Ranking** - We measure the similarity between the vectors using cosine similarity and minimize the pairwise hinge loss. This loss function requires sampling negative samples. This objective is used for TrecQA and WikiQA.
- **Classification** - The concatenation of $[a; b]$ is passed through a standard 2-layer fully-connected layer with h units for classification. The output of the MLP is then passed into a k-class softmax layer and optimized with multi-class cross entropy loss. This loss is adopted for the SciTail (entailment classification task), Twitter (Tweet-Response) and Ubuntu Dialogue datasets.

For both losses, we include a L2 regularization term $\lambda||\theta||_{L2}$. where θ is the trainable model parameters and λ is the weighting term for the regularization term.

4 Experiments

We evaluate our proposed HCRN on four text matching tasks, namely Entailment Classification (premise, hypothesis), Question Answering (question, answer), Tweet Response Prediction (tweet, reply) and Dialogue Prediction (message-reply).

4.1 Experiment 1 - Entailment Classification

Entailment classification is concerned with determining the logical relationship between two sentences, i.e., deciding if the *premise* entails the *hypothesis*.

Experimental Setup

We use the SciTail dataset [Khot *et al.*, 2018], an entailment classification dataset constructed from educational (science) domain, for our evaluation. This dataset comprises $27K$ samples, marked as *entailment* or *neutral*. There are $101, 101$ entail examples and $16, 925$ neutral examples. There are $23K, 1.3K$ and $2K$ pairs for training, development and testing respectively.

[1] An alternative would to be to project directly from the base word embeddings or to use random vectors. Early empirical experiments found these alternatives to perform worse.

Baselines and Implementation

We compare against the benchmarks reported in the actual paper since the splits are identical. The competitors are Enhanced LSTM [Chen *et al.*, 2017] and Decomposable Attention Model [Parikh *et al.*, 2016] and DGEM [Khot *et al.*, 2018], which used knowledge graph triplets to improve the semantic knowledge of the model. The evaluation metric is the accuracy score. For this dataset, we use the two-class classification loss and the *alignment-based* pooling. We use a dimensionality of $d = 100$ for our model. We train all models with the Adam optimizer with an learning rate of 3×10^{-4}. The L2 regularization is set to 10^{-6} and a dropout of $d = 0.8$ is applied to all layers (except the embedding layer). We initialize word embeddings with GloVE $300D$ and keep the embeddings fixed during training. The batch size is set to 64. All parameters are initialized with xavier initialization. We use intra-attention for our model.

Experimental Results

Table 1 reports the results of our evaluation. Firstly, we observe that HCRN achieves the state-of-the-art performance. More notably, we outperform strong baselines such as ESIM and DecompAtt by a large margin ($\approx 10\%$). Additionally, performance gains over DGEM, which uses external knowledge, is significant ($\approx 3\%$). This ascertains the effectiveness of our proposed HCRN model.

Model	Dev	Test
Majority	63.3	60.3
Ngram	65.0	70.6
ESIM [Chen *et al.*, 2017]	70.5	70.6
DecompAtt [Parikh *et al.*, 2016]	75.4	72.3
DGEM w/o edges	75.1	70.8
DGEM	79.6	77.3
HCRN	79.4	**80.0**

Table 1: Performance evaluation (accuracy scores) on SciTail Entailment Classification dataset.

4.2 Experiment 2 - Question Answer Matching

Given a question, retrieval-based question answering (QA) aims to return a ranked list of candidate answers.

Experimental Setup

We evaluate our proposed approach on two popular and widely adopted benchmarks for retrieval-based QA, i.e., WikiQA [Yang *et al.*, 2015] and TrecQA [Wang *et al.*, 2007]. WikiQA comprises $5.9K$ training pairs and $1.1K/1.4K$ development/testing pairs. On the other hand, TrecQA comprises $53K$ pairs for training and $1.1K/1.5K$ pairs for development and testing.

Baselines and Implementation

We compare against a wide range of competitive baselines including AP-BiLSTM [Santos *et al.*, 2016], L.D.C [Wang *et al.*, 2016b], MP-CNN + NCE [Rao *et al.*, 2016] and BiMPM [Wang *et al.*, 2017]. Notably, AP-BiLSTM can be regarded as the key ablation baseline. We use the *alignment-pooling* co-attention and *ranking* loss with a margin λ of 0.1. The number of negative samples are 4 and 6 for WikiQA and TrecQA respectively. We adopt the mix sampling approach in [Rao *et al.*, 2016]. For WikiQA, we use the Adadelta optimizer with a learning rate of 0.1 for WikiQA and 0.2 for TrecQA. Learning rate is decayed at a rate of 0.96 every 10000 steps. The batch size is 100. Sequences are dynamically padded to the batch-wise maximum length. We use the $300D$ GloVe embeddings. All parameters are initialized with Gaussian distributions with zero mean and standard deviation of 0.01. We apply a dropout of 0.9 to all layers.

Experimental Results

Table 2 reports the results on TrecQA and WikiQA.

	TrecQA		WikiQA	
Model	MAP	MRR	MAP	MRR
AP-BiLSTM	0.753	0.851	0.689	0.696
L.D.C	0.771	0.845	0.706	0.723
HyperQA	0.784	0.865	0.712	0.727
MPCNN + NCE	0.801	0.877	0.701	0.718
BiMPM	0.802	**0.899**	0.718	0.731
HCRN	**0.805**	0.895	**0.743**	**0.756**

Table 2: Performance comparison on TrecQA and WikiQA.

Table 2 reports the results on TrecQA and WikiQA. HCRN achieves very competitive performance on TrecQA and outperforms all baselines on WikiQA. HCRN outperforms AP-BiLSTM, a strong co-attentional baseline by a significant margin, i.e., $\approx 4\%$ on TrecQA and $\approx 5\%$ on WikiQA.

4.3 Experiment 3 - Customer Support on Twitter

This experiment is concerned with predicting an appropriate reply given a tweet.

Experimental Setup

We utilize a customer support dataset obtained from Kaggle[2]. This dataset contains tweet-response pairs of tweets to famous brands and their replies. For each Tweet-Reply pair, we randomly selected *four* tweets as negative samples that originate from the same brand. The dataset is split into $8 : 1 : 1$ train-dev-test split. There are $33K$ training samples, $4K$ development samples and $4K$ testing samples. The evaluation metrics for this task are MRR (Mean reciprocal rank) and Precision@1 (accuracy).

Baselines and Implementation

Unlike previous datasets, there are no published works on this dataset. As such, we implement the baselines ourselves. We implement standard baselines such as (1) CBOW (sum embeddings) into a 2 layer MLP with ReLU activations, (2) standard LSTM and CNN models and (3) LSTM and CNN with standard Co-Attention (AP-CNN and AP-LSTM). All attention models utilize *extractive max-pooling* and minimize the binary cross entropy loss. We set all LSTM dimensions to $d = 100$ and the number of CNN filters is 100. The CNN filter width is set to 3. We train all models with the Adam optimizer with 3×10^{-4} learning rate and batch size of 64. The maximum sequence length is set to 30.

[2] https://www.kaggle.com/soaxelbrooke/customer-support-on-twitter

Experimental Results

Model	MRR	P@1
CBOW + MLP	65.8	44.2
Vanilla LSTM	65.2	43.1
Vanilla CNN	68.7	48.0
AP-CNN	69.1	48.7
AP-LSTM	72.4	53.9
HCRN	**73.4**	**55.2**

Table 3: Performance Comparison on Twitter dataset.

Table 3 reports the results of our experiments on the Twitter dataset. HCRN outperforms both AP-LSTM and AP-CNN. Notably, AP-LSTM is the real-valued counterpart of HCRN. As such, this ablation serves as a direct comparison between complex-valued and real-valued co-attention. We observe that HCRN outperforms AP-LSTM by $\approx 1\% - 1.3\%$ in terms of MRR and P@1.

4.4 Experiment 4 - Dialogue Prediction

In this task, the goal is to match a message with replies.

Experimental Setup

We utilize the large and well-known large-scale Ubuntu Dialogue Corpus (UDC) [Lowe *et al.*, 2015]. Following [Wu *et al.*, 2016], the task mainly utilizes the last two utterances in each conversation, predicting if the latter follows the former. We use the same testing splits are provided by Xu et al. [Xu *et al.*, 2016]. The training set comprises **one million** message-response pairs at a $1:1$ positive-negative ratio. The development and testing sets have a $9:1$ ratio. Following [Wu *et al.*, 2016; Xu *et al.*, 2016], we use the evaluation metrics of recall$@k$ ($R_n@K$) which indicates whether the ground truth exists in the top k results from n candidates.

Baselines and Implementation

We compare against a large number of competitive baselines, e.g., MLP, DeepMatch [Lu and Li, 2013], ARC-I / ARC-II [Hu *et al.*, 2014], CNTN [Qiu and Huang, 2015], MatchPyramid [Pang *et al.*, 2016], LSTM, Attentive Pooling LSTM [Santos *et al.*, 2016], MV-LSTM [Wan *et al.*, 2016a] and finally the state-of-the-art Knowledge Enhanced Hybrid Neural Network (KEHNN) [Wu *et al.*, 2016]. Since testing splits are the same, we report the results directly from [Wu *et al.*, 2016]. Following the competitors and for fair comparison, we use a dimensionality of $d = 100$ for the recurrent model. We minimize the classification loss. The Adam optimizer with learning rate of 3×10^{-4} is used. Word embeddings are initialized with GloVE $300D$ and not fine-tuned. Sequence lengths are padded to a maximum of 50 tokens. The batch size is 256 and L2 regularizartion is 10^{-6}. All parameters are initialized with xavier initialization.

Experimental Results

Table 4 reports the experimental results on Ubuntu dialogue corpus. Our proposed HCRN achieves state-of-the-art performance, outperforming a significant number of well-established and competitive baselines. Performance gains over KEHNN are $\approx\ 5\%$ across all metrics while performance gain over AP-LSTM (ablation baseline) ranges from $5\% - 11\%$ on different metrics.

Model	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
MLP	0.651	0.256	0.38	0.703
DeepMatch	0.593	0.345	0.376	0.693
ARC-I	0.665	0.221	0.360	0.684
ARC-II	0.736	0.380	0.534	0.777
CNTN	0.743	0.349	0.512	0.797
MatchPyramid	0.743	0.420	0.554	0.786
LSTM	0.725	0.361	0.494	0.801
AP-LSTM	0.758	0.381	0.545	0.801
MV-LSTM	0.767	0.410	0.565	0.800
KEHNN	0.786	0.460	0.591	0.819
HCRN	**0.816**	**0.508**	**0.656**	**0.863**

Table 4: Performance Comparison on Ubuntu Dialogue Corpus.

4.5 Ablation Study

In order to study the effectiveness of the complex-valued co-attention mechanism, we report scores on three datasets using identical model architectures but only varying the co-attention mechanism. We compare between Complex (bilinear), Complex and Real (Vanilla). Additionally, we also report scores with and without the intra-attention layer.

Dataset	Scitail	Twitter	WikiQA
Complex (Bilinear)	79.5	**73.42/55.24**	0.738/0.746
Complex	77.7	73.22/55.12	0.711/0.723
Real / Vanilla	77.0	72.53/54.07	0.704/0.715
+ With Intra	**80.0**	73.00/54.98	**0.743/0.756**

Table 5: Ablation study on three datasets. Scores in boldface are the best scores

Table 5 reports the ablation results on three datasets. We observe that complex-valued co-attention can improve the performance over real-valued co-attention. Moreover, adding the complex bilinear scoring further improves performance. Finally, the effect of Hermitian intra attention improves performance on WikiQA and Scitail but not on Twitter datasets.

5 Conclusion

We proposed a conceptually simple but highly effective co-attention mechanism for text matching. Our novel approach exploits computation in complex vector space, enabling (1) a sense of word-level directionality, and (2) enhanced representation learning by leveraging complex vector spaces. We demonstrate the effectiveness of our approach on five benchmark datasets and in four different domains. Comparisons against standard co-attention and attention models show that complex-valued co-attention can lead to considerable improvements in performance. Overall, our proposed Hermitian Co-Attention Recurrent Network (HCRN) achieves the state-of-the-art on all datasets.

References

[Arjovsky *et al.*, 2016] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, 2016.

[Chen *et al.*, 2017] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of ACL*, 2017.

[Danihelka *et al.*, 2016] Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, and Alex Graves. Associative long short-term memory. In *Proceedings of ICML*, 2016.

[He *et al.*, 2015] Hua He, Kevin Gimpel, and Jimmy J. Lin. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of EMNLP*, 2015.

[Hu *et al.*, 2014] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS 2014*, 2014.

[Khot *et al.*, 2018] Tushar Khot, Ashish Sabharwal, and Peter Clark. Scitail: A textual entailment dataset from science question answering. In *AAAI*, 2018.

[Lowe *et al.*, 2015] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*, 2015.

[Lu and Li, 2013] Zhengdong Lu and Hang Li. A deep architecture for matching short texts. In *NIPS*, 2013.

[Pang *et al.*, 2016] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. 2016.

[Parikh *et al.*, 2016] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of EMNLP*, 2016.

[Plate, 1995] Tony A. Plate. Holographic reduced representations. *IEEE Trans. Neural Networks*, 6(3), 1995.

[Qiu and Huang, 2015] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of IJCAI*, 2015.

[Rao *et al.*, 2016] Jinfeng Rao, Hua He, and Jimmy J. Lin. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of CIKM*, 2016.

[Santos *et al.*, 2016] Cícero Nogueira Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *CoRR*, abs/1602.03609, 2016.

[Severyn and Moschitti, 2015] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of SIGIR, 2015*, 2015.

[Shen *et al.*, 2014] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of CIKM*, 2014.

[Tay *et al.*, 2017a] Yi Tay, Minh C. Phan, Anh Tuan Luu, and Siu Cheung Hui. Learning to rank question answer pairs with holographic dual LSTM architecture. In *Proceedings of SIGIR*, 2017.

[Tay *et al.*, 2017b] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*, 2017.

[Tay *et al.*, 2017c] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Cross temporal recurrent networks for ranking question answer pairs. *arXiv preprint arXiv:1711.07656*, 2017.

[Tay *et al.*, 2018a] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of WSDM*, WSDM '18, 2018.

[Tay *et al.*, 2018b] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Multi-pointer co-attention networks for recommendation. *arXiv preprint arXiv:1801.09251*, 2018.

[Trabelsi *et al.*, 2017] Chiheb Trabelsi, Olexa Bilaniuk, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. *arXiv preprint arXiv:1705.09792*, 2017.

[Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, 2016.

[Wan *et al.*, 2016a] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of AAAI*, 2016.

[Wan *et al.*, 2016b] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. Match-srnn: Modeling the recursive matching structure with spatial rnn. *arXiv preprint arXiv:1604.04378*, 2016.

[Wang and Jiang, 2016] Shuohang Wang and Jing Jiang. A compare-aggregate model for matching text sequences. *CoRR*, abs/1611.01747, 2016.

[Wang *et al.*, 2007] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. What is the jeopardy model? A quasi-synchronous grammar for QA. In *Proceedings of EMNLP*, 2007.

[Wang *et al.*, 2016a] Bingning Wang, Kang Liu, and Jun Zhao. Inner attention based recurrent neural networks for answer selection. In *Proceedings of ACL*, 2016.

[Wang *et al.*, 2016b] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019*, 2016.

[Wang *et al.*, 2017] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of IJCAI*, 2017.

[Wu *et al.*, 2016] Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. Knowledge enhanced hybrid neural network for text matching. *arXiv preprint arXiv:1611.04684*, 2016.

[Xiong *et al.*, 2016] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604, 2016.

[Xu *et al.*, 2016] Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. *arXiv preprint arXiv:1605.05110*, 2016.

[Yang *et al.*, 2015] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*, 2015.

[Zhang *et al.*, 2018] Peng Zhang, Jiabin Niu, Zhan Su, Benyou Wang, Liqun Ma, and Dawei Song. End-to-end quantum-like language models with application to question answering. 2018.