Hand in Glove: Deep Feature Fusion Network Architectures for Answer Quality Prediction in Community Question Answering

Sai Praneeth Suggu[⋆] **Kushwanth N. Goutham**[⋆]
Manoj K. Chinnakotla[†] **Manish Shrivastava**[⋆]

[⋆]IIIT Hyderabad, India
{suggusai.praneeth,kushwanth.naga}@research.iiit.ac.in
m.shrivastava@iiit.ac.in

[†]Microsoft, India
manojc@microsoft.com

Abstract

Community Question Answering (cQA) forums have become a popular medium for soliciting answers to specific user questions from experts and experienced users in a given topic. However, for a given question, users sometimes have to sift through a large number of low-quality or irrelevant answers to find out the answer which satisfies their information need. To alleviate this, the problem of Answer Quality Prediction (AQP) aims to predict the quality of an answer posted in response to a forum question. Current AQP systems either learn models using - a) various hand-crafted features (HCF) or b) Deep Learning (DL) techniques which automatically learn the feature representations.

In this paper, we propose a novel approach for AQP known as - *"Deep Feature Fusion Network (DFFN)"* which combines the advantages of both hand-crafted features and deep learning based systems. Given a question-answer pair along with its metadata, a DFFN architecture independently - a) learns features using the Deep Neural Network (DNN) and b) computes hand-crafted features leveraging various external resources and then *combines them* using a fully connected neural network trained to predict the quality of the given answer. DFFN is an end-end differentiable model and trained as a single system. We propose two different DFFN architectures which vary mainly in the way they model the input question/answer pair - a) DFFN-CNN which uses a Convolutional Neural Network (CNN) and b) DFFN-BLNA which uses a Bi-directional LSTM with Neural Attention (BLNA). Both these proposed variants of DFFN (DFFN-CNN and DFFN-BLNA) achieve *state-of-the-art performance* on the standard SemEval-2015 and SemEval-2016 benchmark datasets and outperforms baseline approaches which individually employ either HCF or DL based techniques alone.

1 Introduction

Community Question Answering (cQA) forums (such as *Yahoo! Answers*, *Stack Overflow*, *etc.*) have become a popular medium for many internet users to get precise answers or opinions to their questions from experts or other experienced users in the topic. Such forums are usually open, allowing any user to respond to a given question. As a result, for a question posed by the user, the quality of response often varies a lot - ranging from highly precise and detailed answers given by authentic users to highly imprecise or non-comprehensible one-word or single line answers posted by spammy and other non-serious users. This severely hampers the effectiveness of cQA forums since users will have to sift through a large number of irrelevant posts to find the answer which satisfies their information need. To alleviate this problem, cQA forums often include feedback mechanisms such as votes, ratings *etc.* for evaluating the quality of answers and users. Such user feedback could also be used as signals for ranking multiple answers for given a question. However, popularity based signals (votes, ratings) are often prone to spam

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1429–1440, Osaka, Japan, December 11-17 2016.

due to users who may artificially inflate their ratings, votes with the help of other users whom they know. To overcome the above problems, recent approaches (Tran et al., 2015; Hou et al., 2015; Nicosia et al., 2015; Yi et al., 2015; Wang et al., 2009; Zhou et al., 2015; Severyn and Moschitti, 2015; Yu et al., 2014; Filice et al., 2016; Barrón-Cedeño et al., 2016) have focused on automatically ranking answers for a given question based on their quality.

The problem of answer quality prediction is defined as follows: Given a question Q and its set of community answers $C = \{A_1, A_2, \ldots, A_n\}$, rate the answers corresponding to their quality. The cQA tasks of SemEval-2015 (Task A) (Nakov et al., 2015) and SemEval-2016 (Task A) (Nakov et al., 2016) provide a universal benchmark dataset for evaluating research on this problem. In SemEval-2015, the answers are to be rated as {*good,potentially useful or bad*} and in SemEval-2016, the answers are to be rated as either {*good or bad*}.

Recent approaches for answer quality prediction can be categorized into - a) Hand-crafted Feature (HCF) based approaches (Tran et al., 2015; Hou et al., 2015; Nicosia et al., 2015; Yi et al., 2015; Wang et al., 2009; Filice et al., 2016; Barrón-Cedeño et al., 2016) or b) Deep Learning (DL) based approaches (Zhou et al., 2015; Severyn and Moschitti, 2015; Tymoshenko et al., 2016; Yu et al., 2014). HCF based approaches mainly rely on capturing various semantic and syntactic similarities between the question and answer and behavior of users using manual feature engineering. For computing these similarities, recent approaches have also leveraged external knowledge resources such as WordNet and other text corpora. DL based approaches, on the other hand, automatically learn the feature representations while learning the target quality scoring function. As a result, they are language-agnostic and don't require feature engineering or any external resources except for a large training corpus.

In this paper, we propose "Deep Feature Fusion Network (DFFN)" - a novel approach which combines HCF and DL based approaches. The DFFN architecture is designed as a Deep Neural Network (DNN) which takes the question, answer and their metadata as inputs and predicts the quality of answer as output. In the above architecture, HCF is also introduced separately as inputs into the overall DNN. We propose two different architectures of DFFN which mainly differ in the way they model the input question, answer pair - a) Convolutional Neural Network Model (DFFN-CNN) which employs a Convolutional Neural Network (CNN) to model the input question/answer pair and b) Bi-directional Long Short-Term Memory (LSTM) Network Model with Neural Attention (DFFN-BLNA) which uses a Bi-Directional LSTM Model with Neural Attention (BLNA) to model the question/answer pair. DFFN effectively leverages the advantage of both HCF and DL based approaches *i.e.* ability to - a) encode similarities between question-answer pair using external knowledge resources such as Wikipedia, Anchor Text information from Google Cross-Lingual Dictionary (GCD) and Clickthrough data and b) automatically learning features relevant to the target function. During training phase, given a question, answer pair along with its metadata and HCF, DFFN automatically learns deep features which are relevant for the target task using CNN or BLNA. Later, DFFN combines these deep features with HCF, which are computed using various external resources, for predicting the quality rating of the answer. The two proposed architectures of DFFN achieve *state-of-the-art performance* on the standard SemEval-2015 and SemEval-2016 benchmark datasets and also perform better than baseline approaches which individually employ either HCF or DL based techniques. In this context, the following are our main contributions:

- We propose a novel approach to combine resource-based hand-crafted and automatically learnt DL features for the answer quality prediction task.
- We also propose two different architectures of combining HCF and DL based features using CNNs and BLNA.
- Using the above novel architectures, we achieve state-of-the-art performance on SemEval 2015 and SemEval 2016 cQA answer quality prediction tasks.

The rest of the paper is organized as follows: Section 2 discusses the related work in this area. Section 3 presents our contribution DFFN in detail. Section 4 discusses our experimental set-up. Section 5 presents our results and finally Section 6 concludes the paper.

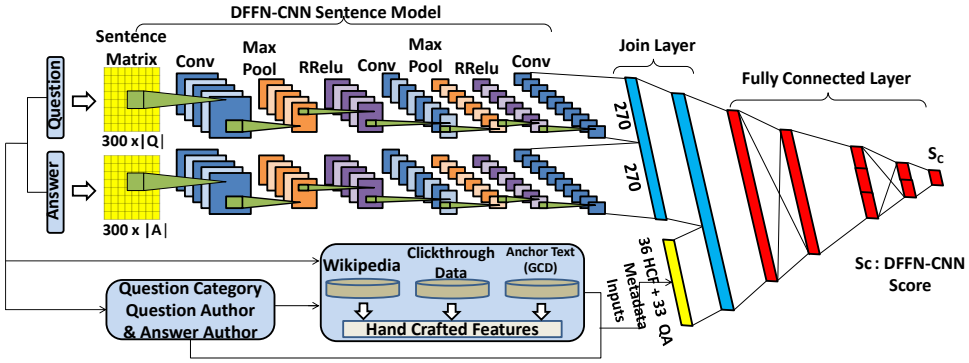

Figure 1: System Architecture of Deep Feature Fusion Network - Convolutional Neural Network with Neural Attention (DFFN-CNN)

2 Related Work

AQP in cQA forums has been researched a lot in the IR community. (Jeon et al., 2006) employ non-textual features such as clicks, print counts, copy counts *etc.* to predict the quality of an answer in a cQA forum. (Liu et al., 2008) investigate a slightly related problem i.e. predicting whether an asker would be satisfied with the answers provided so far to the given question. (Burel et al., 2012) have used a combination of content, user and thread related features for predicting answer quality. (Dalip et al., 2013) propose a learning to rank approach for AQP using eight different groups of features. (Yao et al., 2013; Wang and Manning, 2010) used CRF models with extracted features for AQP. (Li et al., 2015) studied the various factors such as shorter length, authors reputation which lead to a high answer quality rating as rated by peers.

More recently, (Tran et al., 2015) made use of topic models, word vectors and other hand crafted rules to train a SVM classifier for AQP. (Hou et al., 2015) made use of statistics like avg. word length of a sentence (question or answer), sentence length with other hand-crafted features to train an ensemble of classifiers for AQP. (Wang et al., 2009) use Bayesian logistic regression and link prediction models for AQP. (Filice et al., 2016) used kernel based features for AQP.

(Wang and Nyberg, 2015) apply a combination of stacked Bi-Directional LSTMs and keyword matching. (Nicosia et al., 2015) have used lexical similarity between word n-grams, tree kernels, word-embeddings and other hand crafted features for AQP. (Severyn and Moschitti, 2015) used a CNN to automatically learn features for matching short text pairs. (Zhou et al., 2015) used a 2-dimensional CNN to represent a question-answer pair and ranked the representations using a RNN.

Our current work resembles the work of (Wu et al., 2016), in the computer vision community, who employ the idea of combining hand-crafted features and deep features for person re-identification task. However, in our case, the idea of using hand-crafted features is motivated by the availability of large similarity resources such as Wikipedia text, Anchor text of Google Cross-Lingual Dictionary and Click-through data which could be leveraged to infer richer syntactic and semantic similarities between textual elements.

3 Deep Feature Fusion Network (DFFN)

The central idea in a DFFN is to design a Deep Neural Network (DNN) which takes the question (Q), answer (A) and its metadata (MD) as inputs and predicts the quality of an answer as output. DFFN also computes various HCF between Q, A and MD using external resources such as Wikipedia text and Anchor text of Google Cross-Lingual Dictionary (GCD) and Click-through data. These HCFs are also included into the overall DNN as inputs. We propose two different architectures of DFFN depending on the way in which the Q, A pairs are modeled in a NN - a) DFFN-CNN which employs a Convolutional Neural Network (CNN) to model the input question-answer pair and b) DFFN-BLNA which uses a Bi-Directional LSTM Model with Neural Attention (BLNA) to model the question-answer pair. Figures 1,2

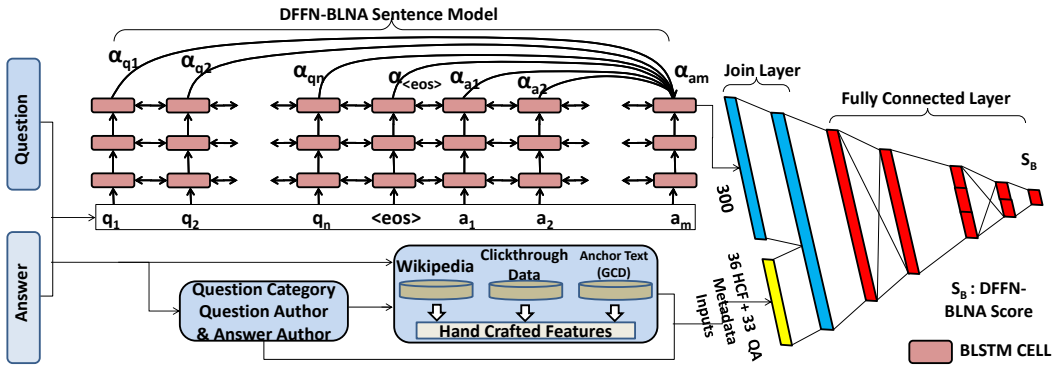

Figure 2: System Architecture of Deep Feature Fusion Network - Bi-directional Long-Short Term Memory Network with Neural Attention (DFFN-BLNA)

depict the architectures of the two proposed variants. Both these variants are end-end differentiable and hence the training is performed end-end.

DFFN-CNN comprises of two parallel CNN based sentence models for the question and answer while DFFN-BLNA has a sequential Bi-directional Long Short-Term Memory Network model with Neural Attention for the question and answer together. Let CNN-FR and BLNA-FR be the deep feature representations generated by using CNN and BLNA respectively. CNN-FR and BLNA-FR are individually joined with HC-FR and metadata and are given as input to a Fully Connected Neural Network (FCNN) which predicts the score representing answer quality. We will now discuss DFFN in detail.

3.1 Sentence Model

DFFN has a sentence model which projects a sentence (question/answer) into the semantic space and learns a good intermediate representation of the given question/answer. The different architectures DFFN-CNN and DFFN-BLNA vary in the way they perform sentence modeling. Here is a brief description of their sentence models:

3.1.1 DFFN - Convolutional Neural Network (DFFN-CNN)

In this architecture, the sentence model is a deep Convolutional Neural Network (CNN). CNN extract features independent of the position in the sentence to create (sub-)sentence representations. CNN consists of sentence matrix and multiple convolutional, pooling and non-linearity layers as in Figure 1 .

Sentence Matrix: The input to the sentence matrix is a vector of words from the sentence (question/answer) $s = [w_1, w_2, ...w_{|s|}]$. We build the sentence matrix by mapping each word w_i in the sentence to its corresponding word embedding in d dimensions. Word embeddings represent similar words by similar vectors and, thus, identify synonyms and other important context words.

We use GLoVE (Pennington et al., 2014) based embeddings of 300 dimensions to map the words in the question and answer. We limit the size of the sentence upto certain threshold. We ignore the words in the sentence after a certain threshold if the length of the sentence is greater than the threshold and pad zeros upto the threshold if the length of the sentence is less than the threshold. The sentence matrix is given as input to the convolutional layer.

Convolution: Convolution is an operation where the feature map (input sentence matrix) and the convolution filter mix together to form a transformed feature map. The convolutional layer extracts patterns i.e, discriminative word sequences that are common in the input train sentences. The convolutional layer is applied on the sentence matrix by convolving a filter with weights $F \in R^{h \times w}$ where h is the filter height and w is the filter width. A filter consisting of a layer of weights is applied to a small patch of sentence matrix to get a single unit as output. The filter is slided across the sentence matrix and the outputs of each patch are combined to get the resultant transformed feature map as the output.

Max Pooling: Max pooling extracts globally most relevant features through local convolution. Max pooling performs a type of non-linear down-sampling. It combines the information and reduce the size

of feature map. It partitions the output of the convolutional layer into small non-overlapping slices and independently operates on every slice by taking the maximum value in each slice as the value in the output of reduced size. We apply max pooling layer on the top of output given by the convolutional layer to extract crucial local features and form a reduced size feature map representation.

Non-Linearity: We use Randomized Leaky Rectified Linear unit (RReLU) (Xu et al., 2015) to learn non-linear decision boundaries. It is a randomized version of leaky ReLU (Xu et al., 2015). RReLU is applied to every element of the output of the max pool layer, thus the resultant feature map will be of the same dimension as the input feature map.The sentence matrix is convolved through multiple convolution, pooling and non-linearity layers to get the feature representations of the question/answer. Using this variation of sentence model, we get the individual feature representations (270 dimensions) of the question and answer. These are concatenated to produce a combined feature representation (540 dimensions).

3.1.2 DFFN - Bi-directional LSTMs with Neural Attention (DFFN-BLNA)

Although, CNN extracts similar patterns on all the patches of the sentence matrix but they do not capture sequential relationships that exists between question-answer pair. LSTMs are memory models which overcome this limitation by feeding the hidden layers from the previous step as an additional input into the next step. In DFFN-BLNA architecture as shown in Figure 2, in stead of a CNN, we use a Bi-directional Long-Short Term Memory (BLSTM) Network with Neural Attention (Bahdanau et al., 2014), for modeling the sentences of question and answer. A question-answer sequence is given as input to BLNA where the sequence is passed through a Bi-directional LSTM Network and the outputs at each step are attended with Neural Attention mechanism. Here, we describe the architecture in more detail.

Long-Short Term Memory (LSTMs) (Hochreiter and Schmidhuber, 1997) are variants of Recurrent Neural Network (RNN) (Werbos, 1990; Rumelhart et al., 1988) architectures which - a) overcome the vanishing and exploding gradients problem of conventional RNNs and b) have the ability to capture long-term dependencies between symbols of a sequence using their gating mechanism which controls information flow. LSTMs only utilize previous context without making use of future context. To overcome this issue, Bidirectional LSTMs (BLSTMs) learn the sequential patterns from both forward and backward directions and then combine information from both directions. The drawback of LSTMs or BLSTMs is that we represent a very long sentence as a single vector which is the output of the last time step. However, using BLSTM with Neural Attention (NA) mechanism, we represent the sequence of vectors as a combined weighted representation vector by selectively attending to the past outputs.

We map the words of the question and answer to their corresponding vector representations using GLoVE embeddings. <eos> is a special symbol used to separate question and answer. A question-answer sequence is generated by concatenating question sentence, <eos> and answer sentence. A BLSTM has two LSTMs that read the QA sequence in both forward and backward directions. At each time step, the output vector is generated by combining the output vectors of two LSTMs, thereby allowing it to consider the contextual information across the entire question-answer sequence i.e. both the question and answer sentence. The neural attention mechanism represents sequence of vectors as a combined weighted representation vector by selectively attending to the past outputs. Thus at each time step, DFFN-BLNA considers the whole context of question and answer and adaptively attends to the subset of past outputs of the BLSTM Network which contributes in better modeling the similarity between question and answer.

Let $Q = [q_1, q_2, ..., q_n]$ be a question of length n and $A = [a_1, a_2, ..., a_m]$ be an answer of length m, then total number of steps in BLSTM will be n+m+1 (additional step is for <eos> after the question). Let $Y = [y_1, y_2, ...y_n, y_{<eos>}, y_{n+2}, y_{n+3}.....y_{n+m+1}]$ be output of BLSTM Network without attention. The combined weighted vector representation c generated using attention mechanism is computed as

$$c = \sum_{i=1}^{n} \alpha_i y_i + (\alpha_{n+1} y_{n+1}) + \sum_{i=n+2}^{m} \alpha_i y_i \tag{1}$$

and $a(y_i, y_k)$ is a latent alignment model which outputs higher score if y_i is useful in capturing the

similarity between question answer pair. where

$$\alpha_i = \frac{\exp(a(y_i, y_{n+m+1}))}{\sum_{j=1}^{n+m+1} \exp(a(y_j, y_{n+m+1}))} \tag{2}$$

Using this variation of sentence model, we generate a combined feature representation of question and answer (300 dimensions). In each of the above architectures, the feature representation derived from the sentence model is combined with the hand crafted features and metadata and is given as input to the Fully Connected Neural Network. We describe these in detail in the following subsections.

3.2 Hand Crafted Features (HCF)

The question and answer text usually consists of several Named Entities (NEs) and concepts along with their various variants. For example, the cricketer *Sachin Tendulkar* could be referred to as *Sachin*, *Tendulkar*, *The Little Master etc.* Such variants are hard to capture using CNN or BLNA based features alone. Hence, we make use of resources such as Wikipedia text, Anchor text of Google Cross-Lingual Dictionary (GCD), Named Entity Recognizers (NER) and Clickthrough data to come up with hand-crafted features which can capture such rich similarities. We also observe that user behavior and specific patterns on metadata and question-answer text are useful. We use these features to compute individual similarity scores between question and answer and combine these scores as Hand Crafted Features to give them as input to the Fully Connected Neural Network. We describe the details of the features below:

3.2.1 Wikipedia Based Features

In this section, we describe the similarity features which are computed by using Wikipedia as a resource.

TagMe Similarity: We extract TagMe concepts from the question and answer by mapping the question and answer to their corresponding Wikipedia page titles using TagMe (Ferragina and Scaiella, 2010). TagMe identifies meaningful substrings in an unstructured text and links them to their relevant wikipedia pages. We compute the similarity between two TagMe concepts using WikiMiner (Milne and Witten, 2013). WikiMiner computes similarity between two wikipedia pages based on the number of common inlinks and outlinks between them. Similarity between question and answer, represented by TagMe concepts, using WikiMiner is computed as the mean average of the similarity between pairs of TagMe concepts (one each from the question and the answer) as in Equation 3

$$qa_{sim} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} sim(c_i, c_j)}{nm} \tag{3}$$

where qa_{sim} is the similarity between question and answer based on TagMe Similarity n, m are the number of TagMe concepts in the question and answer respectively, c_i, c_j are the i^{th} and j^{th} TagMe concepts in the question and answer respectively, $sim(c_i, c_j)$ is the similarity between c_i and c_j calculated using WikiMiner.

Named Entities Similarity: We extract Named Entities from the question and answer, using Stanford CoreNLP NER Tagger (Toutanova et al., 2003) and compute the similarity between two Named Entities using a Google Cross-Lingual Dictionary(GCD) based similarity feature. The GCD based similarity between two Named Entities is computed as the ratio of number of wikipedia documents in which these two named entities co-occur in the top k retrieved documents when queried on GCD. Similarity between question and answer represented by Named Entities is calculated as in Equation 3 where we use Named Entities instead of TagMe concepts and GCD based similarity feature instead of WikiMiner to calculate the similarity between two Named Entities.

3.2.2 AnchorText based Features:

Google Cross-Lingual Dictionary (GCD) (Spitkovsky and Chang, 2012) is a string to concept mapping on the vast link structure of the web, created using anchor text from various pages across the web. A concept is an individual Wikipedia document. The text strings are the anchor texts that refer to these concepts. Thus, each anchor text string represents a concept.

We extract common and proper nouns from the question and answer using Stanford CoreNLP POS Tagger (Toutanova et al., 2003) and query them individually on GCD anchor texts to get top ten unique concepts related to question and answer. We calculate the similarity between two GCD concepts using WikiMiner. The similarity between question and answer represented by GCD concepts is calculated as in Equation 3 where we use GCD concepts instead of TagMe concepts.

3.2.3 Clickthrough Features

Sent2Vec Similarity: Sent2Vec maps a pair of short texts to a pair of feature vectors in a continuous, low-dimensional space. Sent2Vec performs the mapping using the Deep Structured Semantic Model (DSSM) built using Clickthrough data (Huang et al., 2013), or the DSSM with convolutional-pooling structure (CDSSM) (Gao et al., 2014; Shen et al., 2014).

We map the question and answer to vectors using both DSSM and CDSSM. We compute the Sent2Vec DSSM similarity between the question and answer as the cosine similarity between the vectors of question and answer obtained by using Sent2Vec performing the mapping of vectors using DSSM. Similarly by using CDSSM instead of DSSM we also compute the Sent2Vec CDSSM similarity between the question and answer.

Paragraph2vec Similarity: Paragraph2Vec(Le and Mikolov, 2014) allows to model vectors for text of any arbitrary length. It learns continuous distributed vector representations for pieces of texts. We train the para2vec model on the training data of the particular tasks only (SemEval'15 and SemEval'16) by treating each question-answer pair as a single document. We train only on the good question-answer pairs from the training data. A good question-answer pair is a pair in which answer is rated as a "good" answer for that question. We map the question and answer to vectors using para2vec and compute the similarity between the question and answer as the cosine similarity between their para2vec vectors.

3.2.4 Metadata Based Features

Author Reputation Score: We observed that the reputation of an answer author, within a forum plays a key role in determining the quality of answer. We capture this through a author reputation feature. We have two reputation features namely Good Reputation and Bad Reputation. Good reputation of an author is computed as the ratio of the number of good answers given by that author to the maximum number of good answers given by any individual author in the entire forum. Similarly, by using the number of bad answers instead of good answers, we compute a score for the bad reputation of an author. In addition, we also compute Good and Bad reputation scores of an author across each question category.

Is Answer Seeker?: We have a boolean feature to represent whether the answer (comment) is written by the person who has asked the question.

Authors' Response Pattern: We compute features based on whether the question author has commented before or after the present answer and if that comment by the question author is a question. Usually, the question author posts comments/questions below an answer if one is not satisfied with the current answer. Similarly we compute features based on whether the answer author has commented before or after the present answer and if that comment by the answer author is a question. Usually, the answer author posts further questions or comments below ones answer to seek additional information regarding the question or explain his answer more briefly. These features capture the behavior.

Miscellaneous: Besides, we extract and add features related to - a) statistics of each question category (number of good, potential and bad answers in that category) b) position of the answer. c) presence of URL, e-mail in the answer d) presence of question marks, exclamation marks in the answer e) boolean features for the presence of various emoticons such as happy (eg: ":)", ":D"), sad (eg: ":(" , ":'(") in the answer. We obtain the similarity scores and together call them as Hand-crafted features (36 dimensions). We join them as a vector and give them as input to Fully Connected Neural Network along with Metadata as described below.

3.3 Metadata Information

We observe that category of the question plays an important role in answer quality prediction as it may be easy to write good answers for some categories and difficult for some. We encode question category,

SemEval 2015			SemEval 2016			
Model	**F1**	**Acc.**	**Model**	**MAP**	**F1**	**Acc.**
DFFN-BLNA	**62.01***	**75.20***	DFFN-BLNA	**83.91***	**66.70***	**77.65***
DFFN-CNN	60.86	74.54	DFFN-CNN	81.77	65.75	76.42
JAIST	57.29	72.67	Kelp	79.19	64.36	75.11
HITSZ-ICRC	56.44	69.43	ConvKN	78.71	63.55	74.95
DFFN-BLNA *w/o* HCF	56.85	70.45	DFFN-BLNA *w/o* HCF	75.12	61.57	73.12
DFFN-CNN *w/o* HCF	56.06	69.79	DFFN-CNN *w/o* HCF	74.38	60.90	71.96
DFFN *w/o* CNN and BLNA	52.83	66.90	DFFN *w/o* CNN and BLNA	71.56	56,46	69.20
ICRC-HIT	53.82	73.18				

Table 1: Overall Results of DFFN on SemEval 15 and 16 datasets. Results marked with a * were found to be statistically significant with respect to the nearest baseline systems i.e top performing systems of SemEval-15 and 16 at 95% confidence level ($\alpha = 0.05$) when tested using paired two-tailed t-test.

question author and answer author using a logarithmic function and give them as input to the Fully Connected Neural Network.

3.4 Fully Connected Neural Network (FCNN)

The vector representations from the sentence model (540 dimensions from DFFN-CNN or 300 dimensions from DFFN-BLNA), the feature representations from HCF (36 dimensions) and direct inputs from Metadata (33 dimensions) are combined to get a single feature vector of 609 dimensions (DFFN-CNN) or 369 dimensions (DFFN-BLNA). This vector is given as input to FCNN consisting of fully connected layers. These layers model various interactions between the features present in the vector and finally output a score predicting the answer quality.

3.5 Training

The parameters of the network are learnt with an objective to maximize the accuracy of prediction given the target categories. For example, in SemEval-2015, the target categories were $\{good, potentially\ useful, bad\}$ and $\{good, bad\}$ in SemEval-2016 . For training, we used the training data provided in the SemEval 2015 (Nakov et al., 2015) and 2016 (Nakov et al., 2016) tasks which consists of question, answer, metadata along with their ideal quality rating. We tuned the DFFN parameters on the corresponding development sets of SemEval 2015 and 2016. We used adagrad (Duchi et al., 2011) and stochastic gradient descent (SGD) for optimization in DFFN-CNN and DFFN-BLNA respectively.

Given an input (p, t) where p is the predicted answer quality score by DFFN-CNN and t is the true label depicting answer quality, we used SmoothL1 loss criterions which is computed as:

$$loss_{smoothl1}(p,t) = \frac{1}{n} \times \begin{cases} 0.5 \times (p-t)^2, & \text{if } |p-t| < 1 \\ |p-t| - 0.5, & \text{if } |p-t| \geq 1 \end{cases}$$

t is 1 for good question-answer pair (answer labeled as *good* for that question) and -1 for bad question answer pair. (answer labeled as *bad* for that question). The model is trained by minimizing the loss function in a batch of size n. For DFFN-BLNA we used Mean Squared Error (MSE) as loss criterion.

4 Experimental Setup

We use the SemEval 2016 (Nakov et al., 2016) and SemEval 2015 (Nakov et al., 2015) datasets for our experiments as they exactly match our problem description. We use standard evaluation metrics - Mean Average Precision (MAP), F1 score and Accuracy. We compare our approach with the top two best performing systems from SemEval 2015 - JAIST (Tran et al., 2015) and HITSZ-ICRC (Hou et al., 2015) both of which use HCF based models. We also compare with ICRC-HIT (Zhou et al., 2015) as it uses a purely DL based model. Similarly, for SemEval 2016, we compare with their corresponding top two

Question and Answer	TL	DC	JA	HI	IH	Comment
Q: Hi friends, I have State Bank of India Debit card. I try to with draw the money to the atm. It's not accepted. Anybody know which bank atm will accept SBI debit card for withdraw the money ? **A:** dear all banks here accept all international (visa/master/diner club/american express) ATM's cards unless you activate international withdrawal from your mother bank in your mother country.	G	G	B	G	B	***TagMe*** links *visa, master, diner club, american express* to their wiki pages and finds out that they all belong to debit/atm/credit card class.
Q: Can anyone plz help me this problem? I need to send a mobile phone to (Jaipur) India. I contacted DHL but they are charging very high. Is there any other company like DHL? Plz specify... **A:** You can send by post office for cheap price (compare to Courier service)	G	G	B	P	P	***GCD*** similarity feature captures that *post office, DHL, courier* are linked to similar pages when they occur as anchor texts.
Q: What softwares are you using for downloading movies? I'm using limewire and utorrent. How about you? **A:** im using azureus client..limewire sucks (lol)	G	G	B	G	B	***TagMe*** links *azureus, limewire, utorrent* to their wiki pages and finds out that they all belong to movie torrent software class.
Q: I saw a little girl running by the streets , and she had a cat attached to her .....is that normal in this country? **A:** I saw a little girl running by the streets , and she had a parent attached to her .....is that normal in this country?	B	P	P	B	B	***Author Reputation*** gives neutral sim. score; author had written very few answers and had almost equal number of Good and Bad answers. ***Question-Authors' Response Pattern*** gives neutral;author has commented before and after this answer. ***Wiki based*** features gave high scores as question and answer are exactly same except for one word.

Table 2: Qualitative Analysis of DFFN-CNN Results with respect to other baseline approaches. **Note: G: Good, B: Bad, P: Potential; DC: DFFN-CNN, JA: JAIST, HI:HITSZ-ICRC IH:ICRC-HIT**

best performing systems - Kelp (Filice et al., 2016) and ConvKN (Barrón-Cedeño et al., 2016) both of which use kernel-based features.

5 Results and Discussion

Table 1 shows the overall results of DFFN-CNN and DFFN-BLNA on SemEval 2015 and SemEval 2016 datasets. Both the architectures perform better than the top systems across all the metrics. The improvement is higher in SemEval 2015 although the task is harder due to lesser training data and more granularity in target labels. We also observe that DFFN-CNN and DFFN-BLNA perform better than CNN (without HCF) or BLNA (without HCF). Also, the model outperforms hand-crafted feature based model (DFFN with HCF but without CNN or BLNA). Overall, the best performing model was found to be DFFN-BLNA as it more closely models and encodes the semantic dependencies in the QA pair. In Table 2, we present the qualitative analysis of DFFN-CNN architecture results with the best performing baselines on SemEval 2015 dataset. In Table 3, we present the qualitative analysis of DFFN-BLNA architecture results with the best performing baselines on SemEval 2016 dataset. Finally, in Table 4, we compare the performance of DFFN-CNN and DFFN-BLNA using some results from the above datasets.

6 Conclusion

We present a novel approach *"Deep Feature Fusion Networks (DFFN)"*, an end-to-end differentiable approach which combines HCF features into CNN and BLNA models for improving answer quality prediction. DFFN enriches the feature representations learnt through CNN and BLNA by introducing more similarity features computed using external resources such as Wikipedia text, Anchor text of Google Cross-Lingual Dictionary (GCD) and Clickthrough Data. As a result, we show that DFFN achieves *state-of-the-art performance* on the standard SemEval-2015 and SemEval-2016 benchmark datasets and shows better performance than baseline approaches which individually employ either HCF or DL based techniques.

7 Acknowledgement

We acknowledge the support of Crowdfire in the form of an International Travel Grant, to attend this conference.

Question and Answer	TL	DB	KE	CK	Comment
Q: I am very interested to know if there are any expatriate tennis clubs in Doha that anyone can join? I am at a decent standard and would like to play once / twice per week so joining a club would be ideal. If anyone would like a game then please drop me an e-mail and we can arrange something. **A:** We normally play tennis at Khalifa Tennis at least twice a week (Tuesday and Friday); but I would prefer to play for at least 3 times a week or even more. So; if you are interested; I could introduce you to some players so that we could play together.	G	G	B	B	DFFN-BLNA matches with keywords *"tennis","interested,"*. Also due to right intent/response matching.
Q: Has anyone in QL bought any laptop from Jarir bookstore on loan through Qatar Finance company? One more thing;what do you guys think about HP laptops? As i've never bought anything on loan through a bank or a finance company in Qatar **A:** Jarir has an arrangements with QFC that let the customers to purchase laptops and pay back in installments. but their formalities r a bit complex; a lot of documentations required; etc..	G	G	B	G	DFFN-BLNA identifies *Finance Company, Loan* are related to *installments, formalities* . NE *Jarir* matched in both Q&A
Q: My visa is issued and the agency told me it will be going to authenticate it in the embassy here in the philippines how long is the process??? after my visa is stamp what will be the next process?? **A:** If you are going through an agency in the Philippines like what happened to me; it will take at least 1 month of waiting. But in Doha; based on the processing of our PRO in the office; it only takes a week or less. Even for visa renewal; in one week i can have the original I.D.	G	G	G	B	*visa, agency, month, week* co-occur in wiki pages and anchor text of web pages. GCD identified them as similar.
Q: Qtel's settings for mobile internet? I cant seem to access the internet through my Iphone. I've called 111 and they gave me the settings which would be: General - Networks - Cellular Data Network - APN: gprs.qtel Still no signs of getting the internet going. Anyone else have this problem with Iphone? I'm on Shary value pack btw if that info helps. **A:** QTEL is difficult but Vodafone has it's settings on their website. It helped me on my iPhone w/ Vodafone sim.	B	G	B	G	DFFN-BLNA predicts incorrectly due to multiple NE *(Qtel, Iphone)* perfect matches.

Table 3: Qualitative Analysis of DFFN-BLNA Results with respect to other baseline approaches. **Note: G: Good, B: Bad; DB: DFFN-BLNA, KE: Kelp, CK: ConvKN**

Question and Answer	TL	DB	DC	Comment
Q: Hi; my wife was on a visit visa; today; her residency visa was issued; so i went to immigration and paid 500 so there is no need to leave the country and enter again on the residency visa. she has done her medical before for the visit visa extension; do we need to do the medical again for the residency visa? thanks **A:** Hi can u pls. help me ? I just want to know what is the requirements for the family visit visa here in Qatar i want to apply family visit visa for my wife and to my daughter. and also is it true that i can extend the visa up to 6 months? Is there any salary bracket requirements for this visa? I hope u can help me thanks	B	B	G	DFFN-CNN merely compares extent of similarity since keywords like *visa* match while DFFN-BLNA identifies question intent expressed in the sentence.
Q: What is best mall in Doha to buy good furniture? Where are best furniture stores and showrooms. **A:** There are several; my Favorite is Pan Emirates @ Salwa Road.	G	G	B	DFFN-BLNA identified intent of phrase *"There are several"* while DFFN-CNN does not find more explicit matches.
Q: I would like to get some opinion about online job recruitment sites like bayt; gulftalent etc.. Do they really consider the CV's send ? Has anybody got jobs via these online sites ? **A:** They look for key words to match against a given criteria. They have no means of assessing an individual or his/her real skills.	G	B	G	DFFN-CNN finds related keywords in the context such as *"job", "skills", "assessing", "criteria"*.

Table 4: Qualitative Comparison of DFFN-CNN and DFFN-BLNA Results. **Note: G: Good, B: Bad; DB: DFFN-BLNA, DC: DFFN-CNN**

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad Al-Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. 2016. ConvKN at SemEval-2016 Task 3: Answer and Question Selection for Question Answering on Arabic and English Fora. In *SemEval-2016*. ACL.

Grégoire Burel, Yulan He, and Harith Alani. 2012. Automatic Identification of Best Answers in Online Enquiry Communities. In *9th Extended Semantic Web Conference, ESWC 2012*.

Daniel Hasan Dalip, Marcos André Gonçalves, Marco Cristo, and Pavel Calado. 2013. Exploiting User Feedback to Learn to Rank Answers in QA Forums: A Case Study with Stack Overflow. In *SIGIR '13*. ACM.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). CIKM '10. ACM.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. KeLP at SemEval-2016 Task 3: Learning Semantic Relations between Questions and Answers. In *SemEval-2016*. ACL.

Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, Li Deng, and Yelong Shen. 2014. Modeling Interestingness with Deep Neural Networks. EMNLP.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November.

Yongshuai Hou, Cong Tan, Xiaolong Wang, Yaoyun Zhang, Jun Xu, and Qingcai Chen. 2015. HITSZ-ICRC: Exploiting Classification Approach for Answer Selection in Community Question Answering. In *SemEval 2015*. ACL.

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.

Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A Framework to Predict the Quality of Answers with Non-textual Features. In *SIGIR '06*. ACM.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *CoRR*, abs/1405.4053.

Lei Li, Daqing He, Wei Jeng, Spencer Goodwin, and Chengzhi Zhang. 2015. Answer Quality Characteristics and Prediction on an Academic QA Site: A Case Study on ResearchGate. WWW '15 Companion. ACM.

Yandong Liu, Jiang Bian, and Eugene Agichtein. 2008. Predicting Information Seeker Satisfaction in Community Question Answering. In *SIGIR '08*. ACM.

David Milne and Ian H. Witten. 2013. An Open-source Toolkit for Mining Wikipedia. *Artif. Intell.*

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. Semeval-2015 Task 3: Answer Selection in Community Question Answering. In *SemEval '15*. ACL.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 Task 3: Community Question Answering. SemEval '16. ACL.

Massimo Nicosia, Simone Filice, Alberto Barrón-Cedeño, Iman Saleh, Hamdy Mubarak, Wei Gao, Preslav Nakov, Giovanni Da San Martino, Alessandro Moschitti, Kareem Darwish, Lluís Màrquez, Shafiq Joty, and Walid Magdy. 2015. QCRI: Answer Selection for Community Question Answering - Experiments for Arabic and English. In *SemEval 2015*. ACL.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1988. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. SIGIR '15. ACM.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Gregoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. CIKM.

Valentin I. Spitkovsky and Angel X. Chang. 2012. A Cross-Lingual Dictionary for English Wikipedia Concepts. In *LREC'12*. ELRA.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. NAACL '03. ACL.

Quan Hung Tran, Vu Tran, Tu Vu, Minh Nguyen, and Son Bao Pham. 2015. JAIST: Combining multiple features for Answer Selection in Community Question Answering. In *SemEval 2015*. ACL.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2016. Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *Proceedings of NAACL-HLT*, pages 1268–1278.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic Tree-edit Models with Structured Latent Variables for Textual Entailment and Question Answering. COLING '10. ACL.

Di Wang and Eric Nyberg. 2015. A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering. In *ACL 2015*.

Xin-Jing Wang, Xudong Tu, Dan Feng, and Lei Zhang. 2009. Ranking Community Answers by Modeling Question-answer Relationships via Analogical Reasoning. In *SIGIR '09*. ACM.

Paul J. Werbos. 1990. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560. Reprinted in (**?**).

Shangxuan Wu, Ying-Cong Chen, Xiang Li, An-Cong Wu, Jin-Jie You, and Wei-Shi Zheng, editors. 2016. *An Enhanced Deep Feature Representation for Person Re-identification*.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical Evaluation of Rectified Activations in Convolutional Network. *CoRR*, abs/1505.00853.

Xuchen Yao, Benjamin Van Durme, Chris Callison-burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *(NAACL)*.

Liang Yi, JianXiang Wang, and Man Lan. 2015. ECNU: Using Multiple Sources of CQA-based Information for Answers Selection and YES/NO Response Inference. In *SemEval 2015*. ACL.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep Learning for Answer Sentence Selection. *CoRR*, abs/1412.1632.

Xiaoqiang Zhou, Baotian Hu, Jiaxin Lin, Yang xiang, and Xiaolong Wang. 2015. ICRC-HIT: A Deep Learning based Comment Sequence Labeling System for Answer Selection Challenge. In *SemEval 2015*. ACL.