If You Can't Beat Them Join Them: Handcrafted Features Complement Neural Nets for Non-Factoid Answer Reranking

Dasha Bogdanova, Jennifer Foster, Daria Dzendzik and Qun Liu
ADAPT Centre
School of Computing, Dublin City University
Dublin, Ireland
firstname.lastname@adaptcentre.ie

Abstract

We show that a neural approach to the task of non-factoid answer reranking can benefit from the inclusion of tried-and-tested handcrafted features. We present a novel neural network architecture based on a combination of recurrent neural networks that are used to encode questions and answers, and a multilayer perceptron. We show how this approach can be combined with additional features, in particular, the discourse features presented by Jansen et al. (2014). Our neural approach achieves state-of-the-art performance on a public dataset from Yahoo! Answers and its performance is further improved by incorporating the discourse features. Additionally, we present a new dataset of Ask Ubuntu questions where the hybrid approach also achieves good results.

1 Introduction

The task of Question Answering (QA) is arguably one of the oldest tasks in Natural Language Processing (NLP), attracting high levels of interest from both industry and academia. The QA track at the Text Retrieval Evaluation Conference (TREC) was introduced in 1999 and since then has encouraged many research studies by providing a platform for evaluation and making labeled datasets available. However, most research has focused on factoid questions, e.g. the TREC questions *What is the name of the managing director of Apricot Computer?* and *What was the monetary value of the Nobel Prize in 1989?* The TREC QA track organizers took care to "select questions with straightforward, obvious answers" (Voorhees and Tice, 1999) to facilitate manual assessment. In contrast, research on answering non-factoid (NF) questions, such as manner, reason, difference and opinion questions, has been rather piecemeal. This was largely due to the absence of available labeled data for the task. This is changing, however, with the growing popularity of Community Question Answering (CQA) websites, such as Quora,[1] Yahoo! Answers[2] and the Stack Exchange[3] family of forums.

One of the main components of a non-factoid question answering system is the answer reranking module. Given a question, it aims to rearrange the answers in order to boost the community-selected best answer to the top position. Most previous attempts to perform non-factoid answer reranking on CQA data are supervised, feature-based, learning-to-rank approaches (Jansen et al., 2014; Fried et al., 2015; Sharp et al., 2015). These methods represent the candidate answers as meaningful handcrafted features based on syntactic, semantic and discourse parses (Surdeanu et al., 2011; Jansen et al., 2014), web correlation (Surdeanu et al., 2011), and translation probabilities (Fried et al., 2015; Surdeanu et al., 2011). The resulting feature vectors are then passed to a supervised ranking algorithm, such as SVMrank (Joachims, 2006), which ranks the candidates.

There has been a recent shift in Natural Language Processing towards neural approaches involving minimal feature engineering. Several recent studies present purely neural approaches to answer reranking, with most of them focusing on the task of passage-level answer selection (dos Santos et al., 2016; Tan et al., 2015), rather than answer reranking in CQA websites (Bogdanova and Foster, 2016). These neural approaches aim to obviate the need for any feature engineering and instead focus on developing a neural architecture

[1] http://quora.com
[2] http://answers.yahoo.com
[3] http://stackexchange.com

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 121–131,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

that learns the representations and the ranking. However, while it is possible to view a purely neural approach as an alternative to machine learning involving domain knowledge in the form of handcrafted features, there is no reason why the two approaches cannot be applied in tandem. In this paper we show that handcrafted features which encode information about discourse structure can be used to improve the performance of a neural approach to CQA answer reranking.

First, we present a novel neural approach to answer reranking that achieves competitive results on a public dataset of Yahoo! Answers (YA) that was previously introduced by Jansen et al. (2014) and later used in several other studies (Fried et al., 2015; Sharp et al., 2015; Bogdanova and Foster, 2016). Our approach is based on a combination of recurrent neural networks (RNN) and a multilayer perceptron (MLP) that receives the encodings produced by the RNNs and *interaction transformation features* that are based on the outputs of the RNNs and which aim to represent the semantic interaction between the encoded sequences. We also show how this approach can be combined with discourse features previously shown to be beneficial for the task of answer reranking.

The previous best result on the YA dataset – 37.17 P@1 and 56.82 MRR – is reported by Bogdanova and Foster (2016). Our approach achieves similar performance – 37.13 P@1 and 57.56 MRR. In contrast to the (Bogdanova and Foster, 2016) approach, which is also purely neural but requires a large in-domain corpus for pretraining, our model requires only a relatively small training set and no pretraining. The hybrid approach that includes the discourse features outperforms the neural approach on the same dataset and achieves 38.74 P@1 and 58.37 MRR. We also report experiments on a new dataset of Ask Ubuntu[4] questions and answers. The model shows good performance on this dataset too, with the hybrid approach being about 2% more accurate in terms of P@1 than the neural approach on its own. Our error analysis provides insights into the main challenges posed by answer reranking in CQAs. These are the subjective nature of both the questions and the user choice of the best answer.

The main contributions of this paper are as follows: 1) we propose a novel neural approach for non-factoid answer reranking that achieves state-of-the-art performance on a public dataset of Yahoo! Answers; 2) we combine this approach with an approach based on discourse features that was introduced by Jansen et al. (2014), with the hybrid approach outperforming the neural approach and the previous state-of-the-art; 3) we introduce a new dataset of Ask Ubuntu questions and answers.

This paper is organized as follows: an overview of previous work on non-factoid question answering is provided in Section 2, our neural architecture is introduced in Section 3, the discourse features that are incorporated into our neural approach are described in Section 4, the results of our experiments with these new models are presented and analysed in Section 5, and suggestions for further research are provided in Section 6.

2 Related Work

Previous work on supervised non-factoid answer reranking on CQA datasets focused mainly on feature-rich approaches. Surdeanu et al. (2011) show that CQAs such as Yahoo! Answers are a good source of knowledge for non-factoid QA. They employ four types of features in their answer reranking model: (1) similarity features: the similarity between a question and an answer based on the length-normalized BM25 formula (Robertson et al., 1994); (2) translation features: probability of the question being a translation of the answer computed using IBM's Model 1 (Brown et al., 1993); (3) features measuring frequency and density of the question terms in the answer, such as the number of non-stop question words in the answer, the number of non-stop nouns, verbs and adjectives in the answer that do not appear in the question and tree kernel values for question and answer syntactic structures; (4) web correlation features based on Corrected Conditional Probability (Magnini et al., 2002) between the question and the answer. They explore these features both separately and in combination and find that the combination of all four feature types is most beneficial for answer reranking models.

Jansen et al. (2014) describe answer reranking experiments on YA using a diverse range of lexical, syntactic and discourse features. In particular, they show how discourse information can complement distributed lexical semantic information obtained with a skip-gram model (Mikolov et al., 2013). In this paper we use their features (discussed in detail in Section 4) in combination with

[4] http://askubuntu.com

a neural approach. Fried et al. (2015) improve on the lexical semantic models of Jansen et al. (2014) by exploiting indirect associations between words using higher-order models.

Methods based purely on neural models have gained popularity in various areas of NLP in recent years. The main advantage of these models is that they are often able to achieve state-of-the-art results while obviating the need for manual feature engineering. These approaches have been successful in the area of question answering. Several studies proposed models based on convolution neural networks (Severyn and Moschitti, 2015; Tymoshenko et al., 2016; Feng et al., 2015) for answer sentence selection for factoid question answering and models based on combinations of convolutional and recurrent neural networks for the task of passage-level non-factoid answer reranking (Tan et al., 2015; dos Santos et al., 2016). Recurrent neural networks and memory networks were successfully applied to the task of reading comprehension (Xiong et al., 2016; Sukhbaatar et al., 2015; Weston et al., 2015). A simple purely neural approach to non-factoid answer reranking in CQAs was proposed by Bogdanova and Foster (2016). The question-answer pairs are represented with Paragraph Vector (Le and Mikolov, 2014) distributed representations, and a multilayer perceptron is used to estimate the probability of the answer being good for the given question. The approach achieves state-of-the-art results. However, it requires unsupervised pretraining of the Paragraph Vector model on a relatively big in-domain dataset.

Recently, the Wide and Deep learning model for recommendation systems was proposed (Cheng et al., 2016). This model trains a *wide* linear model based on sparse features alongside a deep neural model, thus combining the benefits of memorization provided by the former part and the generalization provided by the latter.

In this paper, we propose a hybrid approach to answer reranking. Similarly to the wide and deep model, it combines traditional feature-based and deep neural approaches. However, in this paper we enhance the neural model with discourse chunk features that were previously found useful for this task. The features are combined with a neural model that consists of two bidirectional RNNs that encode the question and the answer and a multilayer perceptron that receives the neural encodings and the discourse features and makes the final prediction.

3 Learning to rank answers with RNNs and MLP

We illustrate our approach to answer reranking in Figure 1. Following previous research on neural answer reranking (Severyn and Moschitti, 2015; Bogdanova and Foster, 2016), we employ the pointwise approach to ranking, i.e. we cast the ranking task as a classification task. Given a question q and an answer a, we first use two separate bidirectional RNNs[5] to encode the question and the answer. Let $(w_1^q, w_2^q, ..., w_k^q)$ be the sequence of question words and $(w_1^a, w_2^a, ..., w_p^a)$ be the sequence of answer words.[6] The first RNN encodes the sequence of question words into the sequence of context vectors $(h_1^q, h_2^q, ..., h_k^q)$, i.e.

$$f_{RNN}^q(w_i^q, \theta_q) = h_i^q \tag{1}$$

where θ_q denote the trainable parameters of the network. More specifically, the bidirectional RNN consists of two RNNs: the forward RNN that reads the question starting from the first word until the last word and encodes it as a sequence of forward context vectors $(\overrightarrow{h_1^q}, \overrightarrow{h_2^q}, ..., \overrightarrow{h_k^q})$, and the reverse RNN that encodes the question starting from the last word until the first word: $(\overleftarrow{h_k^q}, \overleftarrow{h_{k-1}^q}, ..., \overleftarrow{h_1^q})$. The resulting context vectors are concatenations of the forward and reverse context vectors at each step, i.e. $h_i^q = [\overrightarrow{h_i^q}, \overleftarrow{h_i^q}]$. As the encoded vector representation of the question, we use the concatenation of the context vectors, i.e.

$$enc^q = [h_1^q, ..., h_k^q] \tag{2}$$

The second bidirectional RNN encodes the answer in the same way:

$$f_{RNN}^a(w_i^a, \theta_a) = h_i^a \tag{3}$$

$$enc^a = [h_1^a, ..., h_p^a] \tag{4}$$

where θ_a denote the trainable parameters of the network. We also want to optionally explicitly encode the interaction between the question's context vectors and the answer's context vectors. To

[5] We use an RNN with Gated Recurrent Units (GRU) (Bahdanau et al., 2015). Using an LSTM instead provides similar results.

[6] The questions and answers have to be padded to k and p words respectively.

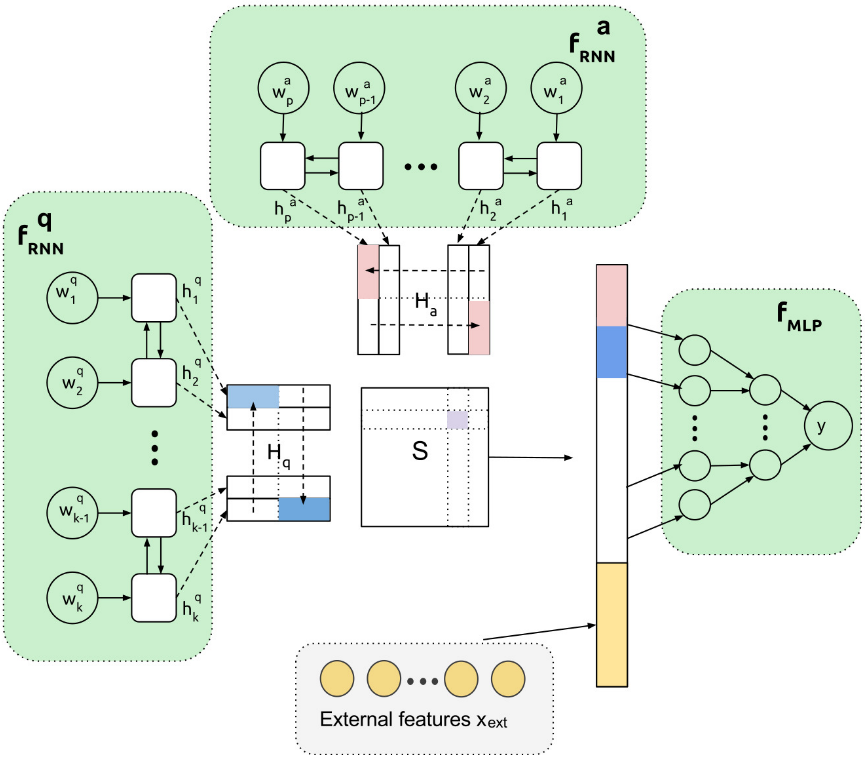

Figure 1: Our model takes a question-answer pair as an input and encodes them using separate RNNs denoted as f^q_{RNN} and f^a_{RNN}. Then a similarity matrix S over the encodings is computed and optionally concatenated with external features x_{ext}, the result is passed to a multilayer perceptron f_{MLP} that outputs the final prediction.

do this we apply the *interaction transformation* to the context vectors. More specifically, let H_q denote the matrix composed of the outputs of the question encoder RNN:

$$H_q = \begin{pmatrix} h^q_{1,1} & h^q_{1,2} & \cdots & h^q_{1,k} \\ h^q_{2,1} & h^q_{2,2} & \cdots & h^q_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ h^q_{d,1} & h^q_{d,2} & \cdots & h^q_{d,k} \end{pmatrix}$$

and H_a denote the matrix composed of the outputs of the answer RNN:

$$H_a = \begin{pmatrix} h^a_{1,1} & h^a_{1,2} & \cdots & h^a_{1,p} \\ h^a_{2,1} & h^a_{2,2} & \cdots & h^a_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ h^a_{d,1} & h^a_{d,2} & \cdots & h^a_{d,p} \end{pmatrix}$$

d is a dimensionality parameter to be experimentally tuned. We calculate the similarity matrix S between H_q and H_a, so that each element s_{ij} of the S matrix is a dot product between the corresponding encodings:

$$s_{ij} = h^q_i \cdot h^a_j$$

The similarity matrix S is unrolled and passed to the multilayer perceptron along with the question and answer encodings. They are optionally concatenated with external features x_{ext}:

$$y = f_{MLP}([S, enc^q, enc^a, x_{ext}], \theta_s) \qquad (5)$$

where θ_s denote the trainable parameters of the network. The network is trained by minimizing cross-entropy:

$$L(y, \theta) = -\bar{y}\log(y) - (1 - \bar{y})\log(1 - y)$$

where θ are all network's parameters, i.e. $\theta_q, \theta_a, \theta_s$ and $\bar{y}$ is the true label:

$$\bar{y} = \begin{cases} 1 & \text{if } a \text{ is the best answer of the question } q \\ 0 & \text{otherwise} \end{cases}$$

4 Discourse Features

Based on the intuition that modelling question-answer structure both within and across sentences could be useful, Jansen et al. (2014) propose an answer reranking model based on discourse features

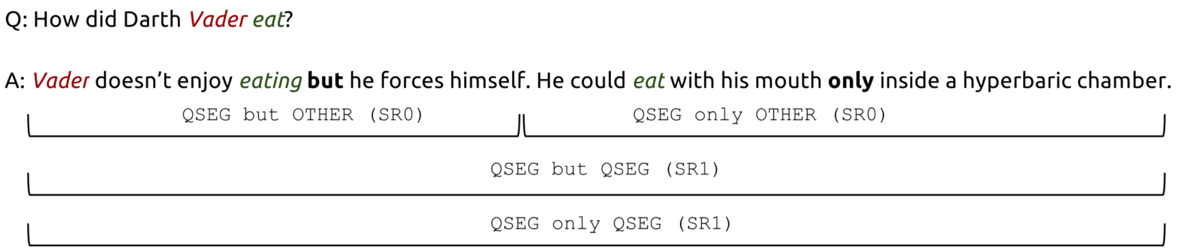
Q: How did Darth Vader eat?
A: Vader doesn't enjoy eating **but** he forces himself. He could eat with his mouth **only** inside a hyperbaric chamber.
QSEG but OTHER (SR0)
QSEG only OTHER (SR0)
QSEG but QSEG (SR1)
QSEG only QSEG (SR1)

Figure 2: Feature generation for the discourse marker model of Jansen et al. (2014): first, the answer is searched for the discourse markers (in **bold**). For each discourse marker, there are several features that represent if there is an overlap (`QSEG`) with the question before and after the discourse marker. The features are extracted for sentence range from 0 (the same range) to 2 (two sentences before and after). .

combined with lexical semantics. We experimentally evaluate these discourse features – added to our model described in Section 3 (the additional features x_{ext}) and on their own. We reuse their discourse marker model (DMM) combined with their lexical semantics model (LS). The DMM model is based on the findings of Marcu (1998), who showed that certain cue phrases indicate boundaries between elementary textual units with sufficient accuracy. These cue phrases are further referred to as discourse markers. For English, these markers include *by, as, because, but, and, for* and *of* – the full list can be found in Appendix B in (Marcu, 1998).

We illustrate the feature extraction process of Jansen et al. (2014) in Figure 2. First, the answer is searched for discourse markers. Each marker divides the text into two arguments: preceding and following the marker. Both arguments are searched for words overlapping with the question. Each feature denotes the discourse marker and whether there is an overlap with the question (`QSEG`) or not (`OTHER`) in the two arguments defined by the marker. The sentence range (`SR`) denotes the length (in sentences) of the marker's arguments. For example, `QSEG by OTHER SR0` means that in the sentence containing the *by* marker there is an overlap with the question before the marker and there is no overlap with the question after the marker. This results in 1384 different features. To assign values to each feature, the similarity between the question and each of the two arguments is computed, and the average similarity is assigned as the value of the feature. Jansen et al. (2014) use cosine similarity over *tf.idf* and over the vector space built with a skip-gram model (Mikolov et al., 2013). Further details can be found in (Jansen et al., 2014).

5 Experiments

5.1 Data

In our experiments, we use two datasets from different CQAs. For comparability, we use the dataset created by Jansen et al. (2014) which contains 10K *how* questions from Yahoo! Answers. 50% of it is used for training, 25% for development and 25% for testing. Each question in this dataset contains at least four user-generated answers. Some examples can be found in Table 1. Further details about this dataset can be found in (Jansen et al., 2014).

To evaluate our approach on a more technical domain, we create a dataset of Ask Ubuntu (AU) questions containing 13K questions, of which 10K are used for training, 0.5K for development and 2.5K for testing. The Ask Ubuntu community is a part of the Stack Exchange family of forums. Forums of this family share the same interface and guidelines. They allow users to post questions and answers and to vote them up and down, resulting in every question and every answer having a score representing the votes it received. The author of the question may select the *best answer* to their question. We create the AU dataset in the same way as the YA dataset was created: for each question, we only rank answers provided in response to this question, and the answer labelled as the best by the question's author is considered to be the correct answer. We make sure that the dataset contains only questions that have at least three user-provided answers and have the best answer selected, and that this answer has a non-negative score. Example questions from this dataset can be

Question: how do you cut onions without crying?

Gold: Use a sharp knife because if the onions are cut cleanly instead of slightly torn (because of a dull knife) they will release less of the chemical that makes you cry. Lighting a candle also helps with this, (...) I hope this helps.

Other Answers:
- Watch a comedy.
- Put onion in the chop blender
- close ur eyes...
- Sprinkle the surrounding area with lemon juice.
- Choose one of the followings after cutting the head and tail of the onion, split in half and peel off the skin. 1. Keep on chopping with your knife 2. Cut in quarters and put in choppers.

Table 1: Example question from the Yahoo! Answers dataset.

Question: Can't shutdown through terminal. When ever i use the following `sudo shutdown now; sudo reboot; sudo shutdown -h` my laptop goes on halt (...) is there something wrong with my installation?

Gold: Try the following code `sudo shutdown -P now` (...) -P Requests that the system be powered off after it has been brought down. -c Cancels a running shutdown. -k Only send out the warning messages and disable logins, do not actually bring the system down.

Other Answers:
- Try `sudo shutdown -h now` command to shutdown quickly.
- Try `init 0` init process shutdown all of the spawned processes/daemons as written in the init files

Table 2: Example question from the Ask Ubuntu dataset.

found in Table 2.

There are significant differences between the two datasets. While the Yahoo! Answers dataset has very short questions (10.8 on average) and relatively long answers (50.5 words), Ask Ubuntu questions can be very long, as they describe non-trivial problems rather than just ask questions. The average length of the Ask Ubuntu questions is 112.14 words, with the average answer being about 95 words long.

5.2 Experimental Setup

Following Jansen et al. (2014) and Fried et al. (2015), we implement two baselines: the baseline that selects an answer randomly and the candidate retrieval (CR) baseline. The CR baseline uses the same scoring as in Jansen et al. (2014): the questions and the candidate answers are represented using *tf-idf* over lemmas; the candidate answers are ranked according to their cosine similarity to the respective question. Additionally, we evaluate the discourse features described in Section 4 alone: we use them as the representation of the question-answer pairs that are then used as the input to a multilayer perceptron with five hidden layers. On the YA dataset, we also compare our results to the ones reported by Jansen et al. (2014) and by Bogdanova and Foster (2016).

The model described in Section 3 is regularized with L2-regularization and dropout. The development sets are used solely for early stopping and hyperparameter selection. We tune the hyperparameters (learning rate, L2 regularization rate, dropout probabilities, dimensionality of the embeddings, the network architecture (the number of hidden layers and units, the use of GRU versus LSTM)) on the development sets. All neural networks use the rectified linear activation function (ReLU). The word embeddings are initialized randomly, no pretrained embeddings are used. We use the software provided by Jansen et al. (2014)[7] to extract the discourse features described in Section 4 and referred to as x_{ext} in Section 3. These discourse features require that word embeddings be trained in order to calculate the similarity. Following Jansen et al. (2014), we train them using the skip-gram model (Mikolov et al., 2013) We use the L6 Yahoo dataset[8] to train the skip-gram model for the YA dataset and the Ask Ubuntu September 2015 data dump for the AU dataset. The neural model described in Section 3 does not require pretraining of word embeddings, the embeddings are used only to extract external discourse features. To evaluate all the models, we use standard implementations of P@1 and mean reciprocal rank (MRR).

5.3 Results

We experimentally evaluate the following models:

[7] http://nlp.sista.arizona.edu/releases/acl2014/

[8] http://webscope.sandbox.yahoo.com/

- **MLP-discourse:** The discourse features are extracted as described in Section 4, an MLP is used to produce the ranking;

- **GRU-MLP:** The system described in Section 3 without the interaction matrix S and any other external features (x_{ext} in Section 3 and in Figure 1);

- **GRU-MLP-Sim:** The system described in Section 3 with the interaction matrix S and no external features;

- **GRU-MLP-Sim-Discourse:** The system described in Section 3 with the interaction matrix S and the discourse features as the external features x_{ext};

Table 3 reports the answer reranking P@1 and MRR of the described models along with the results of the baseline systems. The models were frozen on their best development epoch, the test set had been used neither for model selection nor for parameter tuning.[9]

Table 3 shows that the discourse features on their own with an MLP (MLP-Discourse) outperform the random and the CR baselines for both datasets. They also perform better than the approach of Jansen et al. (2014) who used SVMrank with a linear kernel. This might be due to the ability of the MLP to model non-linear dependencies. Nonetheless, the MLP-Discourse approach performs worse than the approach of Bogdanova and Foster (2016), which is based on distributed representations of documents, which probably capture more information relevant to the task.

The system described in Section 3 with no interaction transformation (only the encodings are passed to the MLP) and without any external features (x_{ext} in Section 3 and in Figure 1), referred to as GRU-MLP, outperforms the CR and the Random baselines and the systems based on the discourse features. However, it performs slightly worse than the approach of (Bogdanova and Foster, 2016). One possible reason is that the latter uses a large corpus for unsupervised pretraining. The GRU-MLP systems does not use any external data, and learns only from the small training set.

Yahoo! Answers

Model	P@1	MRR
Random Baseline	15.74	37.40
CR Baseline	22.63	47.17
Jansen et al. (2014)	30.49	51.89
Bogdanova and Foster (2016)	37.17	56.82
MLP-Discourse	32.72*	53.54*
GRU-MLP	36.12*	56.63*
GRU-MLP-Sim	37.13*	57.56*
GRU-MLP-Sim-Discourse	**38.74***	**58.37***

Ask Ubuntu

Model	P@1	MRR
Random Baseline	26.60	53.64
CR Baseline	35.36	60.17
MLP-Discourse	37.80*	61.75*
GRU-MLP	38.56*	62.53*
GRU-MLP-Sim	39.28*	62.64*
GRU-MLP-Sim-Discourse	**41.40***	**64.42***

Table 3: The systems results versus the baselines. * The improvements over the CR and Random baselines are statistically significant with $p < 0.05$. All significance tests are performed with one-tailed bootstrap resampling with 10,000 iterations.

The system enriched with the interaction matrix, GRU-MLP-Sim, clearly outperforms all the baselines on both datasets, including the MLP-Discourse system. On the YA dataset, the results are better than Jansen et al. (2014) and very similar to Bogdanova and Foster (2016). On the AU dataset the improvement over the CR and the MLP-discourse systems is less remarkable, yet statistically significant. This indicates the benefit of explicitly providing the interaction features to the MLP.

The same approach with the additional discourse features described in Section 4, referred to as GRU-MLP-Sim-Discourse in Table 3, achieves the highest P@1 and MRR on the YA dataset and the AU dataset. Surprisingly, the discourse features are very helpful on the AU dataset which is highly technical, with significant parts of the information represented as commands and code.

Even though the results achieved on both datasets are similar in absolute values, the datasets are very different and the errors might be of a different nature. We provide some insights into the

[9] We report the results obtained with a bidirectional RNN with GRU cell, MLP with 5 hidden layers (with 5120, 2048, 1024, 512, 128 units), batch size 100, learning rate 0.01, weight decay 0.0005, dropout keep probability 0.6, and the word embedding dimensionalities and RNN outputs set to 100. The questions and answers are padded: the lengths are set to 15 words for the question and 100 words for the answer in the YA dataset and 200 and 150 words for the AU dataset.

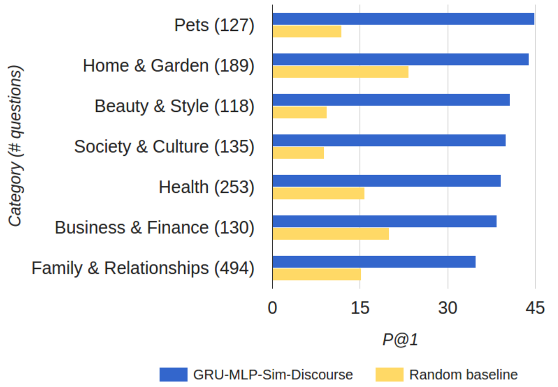

Figure 3: Average P@1 of the GRU-MLP-Sim-Discourse versus the Random baseline on the test questions from most common YA categories.

challenges raised by the two datasets in the next section.

5.4 Error Analysis

By conducting an error analysis on the YA dataset we were able to pinpoint the main causes of error as follows:

1. Despite containing only *how* questions, the dataset contains a large amount of questions asking for an **opinion** or **advice** , e.g. *How should I do my eyes?*, *How do I look?* or *How do you tell your friend you're in love with him?* rather than **information**, e.g. *How do you make homemade lasagna?* and *how do you convert avi to mpg?* About half of the questions where the best system was still performing incorrectly were of the opinion-seeking nature. This is a problem for automatic answer reranking, since the nature of the question makes it very hard to predict the quality of the answers.

2. The choice of the best answer purely relies on the user. Inspection of the data reveals that these user-provided gold labels are not always reliable. In many cases the users tend to select as the best those answers that are most **sympathetic** (see (Q1) in Table 4) or **funny** (see (Q2) and (Q3) in Table 4), rather than the ones providing more useful information.

In order to gain more insights into the reasons behind errors on the YA data, we calculated average P@1 per category.[10] Figure 3 shows average P@1 of the GRU-MLP-Sim-Discourse system versus the Random baseline for the most common categories. From this figure it is clear that the most challenging category for answer reranking is Family & Relationships. This category is also the most frequent in the dataset, with 494 out of 2500 questions belonging to it. Our system achieves about 4% lower P@1 on the questions from Family & Relationships category than on the whole test set, while the random baseline performs as well as on the whole test set (the average number of answers per question in this category does not differ much from the dataset average). The low P@1 on this category is related to the reasons pointed out above: most questions in this category are of an opinion-seeking nature: *How do I know if my boyfriend really loves me?*, *How do I fix my relationship?*, *How do I find someone that loves me?*, making it hard to assess the quality of the answers.

The Ask Ubuntu dataset is rather different. In contrast to the YA dataset, which contains many subjective questions, most Ask Ubuntu questons relate to a complex technology and usually require deep domain knowledge to be answered. Moreover, many questions and answers contain code, screenshots and links to external resources. Reliably reranking such answers based on textual information alone might be an unattainable goal. The technical complexity of the questions can give rise to ambiguity. For instance, in (Q2) in Table 5 it is not clear if the question refers to the metapackage *ubuntu-desktop* or to ubuntu default packages in general. Another potential source of difficulty comes from the fact that the technologies being discussed on Ask Ubuntu change rapidly: some answers selected as best might be outdated (see (Q1) in Table 5).

6 Conclusions and Future Work

In this paper we presented a neural approach to open-domain non-factoid answer reranking. Previous studies in this area have either been feature-based or purely neural approaches that require no manual feature engineering. We show that these two approaches can be successfully combined. We propose a novel neural architecture whereby the question-answer pairs are first encoded using two

[10] We first mapped the low-level categories provided in the dataset to the 26 high-level YA categories. We only consider categories that contained at least 100 questions.

(Q1) How does someone impress a person during a conversation that u are as good as an oxford/harvard grad.?

(Gold) i think you’re chasing down the wrong path. but hell, what do i know?

(Prediction) There are two parts. Understanding your area well, and being creative. The understanding allows you the material for your own opinions to have heft and for you to analyse the opinions of others. After that, it’s just good vocabulary which comes from reading a great deal and speaking with others. Like many other endeavors practice is what makes your performance improve.

(Q2) How to get my mom to stop smoking?

(Gold) Throw a glass of water on her every time she sparks one up

(Prediction) Never nag her. Instead politely insist on your right to stay free of all the risks associated with another person’s
smoking. For example, do not allow her to smoke inside the car, the house or anywhere near you (...)

(Q3) How do i hip hop dance??!?!?

(Gold) Basically, you shake what your mother gave you.

(Prediction) Listen to previous freestyle flows and battles by great artists (...) Understand the techniques those artists use to flow and battle (...)

Table 4: Example incorrect predictions of the system on the Yahoo! Answers dataset.

(Q1) How do I add the kernel PPA? I can get Ubuntu mainline kernels from this kernel PPA - is there a way to add it to my repository list the same as regular Launchpad PPAs?

(Gold) Warning : This answer is outdated. As of writing this warning (6.10.2013) the kernel-ppa used here is no longer updated. Please disregard this answer. `sudo apt-add-repository ppa:kernel-ppa/ppa sudo apt-get update sudo apt-get install PACKAGENAME`

(Prediction) Since the kernel ppa is not really maintained anymore, here’s a semi-automatic script: `https://github.com/medigeek/kmp-downloader`

(Q2) Which language is ubuntu-desktop mostly coded in? I heard it is Python

(Gold) Poked around in Launchpad: ubuntu-desktop to and browsed the source for a few mins. It appears to be a mix of Python and shell scripts.

(Prediction) I think the question referred to the language used to write the applications running on the default installation. It’s hard to say which language is used the most, but i would guess C or C++. This is just a guess and since all languages are pretty equal in terms of outcome, it doesn’t really matter.

Table 5: Example incorrect predictions of the system on the Ask Ubuntu dataset.

recurrent neural networks, then the interaction matrix is calculated, concatenated with external features, and passed as an input to a multilayer perceptron. As external features, we evaluate the discourse features that were found useful for this task by Jansen et al. (2014). The combined approach achieves new state-of-the-art results on two CQA datasets.

However, despite these encouraging results, the P@1 is still below 40%. As the error analysis shows, this is due to the nature of the dataset: the user choice of the best answer is not always reliable and the questions are often seeking opinions rather than information. The ceiling for this task could be very low. Manual annotation of CQA data might help in determining the upper bound.

Future work should aim to create more reliable gold standards for this task. As we show in this paper, the CQAs contain various types of question: some of which are seeking information and some not. Existing corpora of opinion questions, such as the OpQA corpus (Stoyanov et al., 2005), could be used in future research to distinguish those from the information-seeking questions. Another possible direction for future work is in combining the neural approach with other external features, such as features based on web correlation between the question and the answer, and similarities between their syntactic structures.

Acknowledgements

This research is supported by Science Foundation Ireland through the CNGL Programme (Grant 12/CE/I2267) in the ADAPT Centre for Digital Content Technology. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *5th International Conference on Learning Representations 2015*.

Dasha Bogdanova and Jennifer Foster. 2016. This is how we do it: Answer reranking for open-domain how questions with paragraph vectors and minimal feature engineering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1290–1295, San Diego, California. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. *arXiv:1606.07792*.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv:1602.03609*.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 813–820. IEEE.

Daniel Fried, Peter Jansen, Gustave Hahn-Powell, Mihai Surdeanu, and Peter Clark. 2015. Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association for Computational Linguistics*, 3:197–210.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 977–986, Baltimore, Maryland, June. Association for Computational Linguistics.

Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002. Is it the right answer?: exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 425–432. Association for Computational Linguistics.

Daniel C. Marcu. 1998. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. Ph.D. thesis, Toronto, Ont., Canada, Canada.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Proceedings of the 3rd Text REtrieval Conference*, pages 109–126.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382.

Rebecca Sharp, Peter Jansen, Mihai Surdeanu, and Peter Clark. 2015. Spinning straw into gold: Using free text to train monolingual alignment models for non-factoid question answering. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 231–237, Denver, Colorado, May–June. Association for Computational Linguistics.

Veselin Stoyanov, Claire Cardie, and Janyce Wiebe. 2005. Multi-perspective question answering using the opqa corpus. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 923–930. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Comput. Linguist.*, 37(2):351–383, June.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv:1511.04108*.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2016. Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 1268–1278, San Diego, California, June. Association for Computational Linguistics.

Ellen M. Voorhees and Dawn M. Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. *arXiv:1603.01417*.