This is how we do it: Answer Reranking for Open-domain How Questions with Paragraph Vectors and Minimal Feature Engineering

Dasha Bogdanova and Jennifer Foster
ADAPT Centre
School of Computing, Dublin City University
Dublin, Ireland
{dbogdanova,jfoster}@computing.dcu.ie

Abstract

We present a simple yet powerful approach to non-factoid answer reranking whereby question-answer pairs are represented by concatenated distributed representation vectors and a multilayer perceptron is used to compute the score for an answer. Despite its simplicity, our approach achieves state-of-the-art performance on a public dataset of *How* questions, outperforming systems which employ sophisticated feature sets. We attribute this good performance to the use of paragraph instead of word vector representations and to the use of suitable data for training these representations.

1 Introduction

In contrast to factoid question answering (QA), non-factoid QA is concerned with questions whose answer is not easily expressed as an entity or list of entities and can instead be quite complex – compare, for example, the factoid question *Who is the secretary general of the UN?* with the non-factoid manner question *How is the secretary general of the UN chosen?* A significant amount of research has been carried out on factoid QA, with non-factoid questions receiving less attention. This is changing, however, with the popularity of community-based question answering (CQA) sites such as Yahoo! Answers[1], Quora[2] and the StackExchange[3] family of forums. The ability of users to vote for their favourite answer makes these sites a valuable source of training data for open-domain non-factoid QA systems.

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

In this paper, we present a neural approach to open-domain non-factoid QA, focusing on the subtask of answer reranking, i.e. given a list of candidate answers to a question, order the answers according to their relevance to the question. We test our approach on the Yahoo! Answers dataset of manner or *How* questions introduced by Jansen et al. (2014), who describe answer reranking experiments on this dataset using a diverse range of features incorporating syntax, lexical semantics and discourse. In particular, they show how discourse information (obtained either via a discourse parser or using shallow techniques based on discourse markers) can complement distributed lexical semantic information. Sharp et al. (2015) show how discourse structure can be used to generate artificial question-answer training pairs from documents, and test their approach on the same dataset. The best performance on this dataset – 33.01 P@1 and 53.96 MRR – is reported by Fried et al. (2015) who improve on the lexical semantic models of Jansen et al. (2014) by exploiting indirect associations between words using higher-order models.

In contrast, our approach is very simple and requires no feature engineering. Question-answer pairs are represented by concatenated distributed representation vectors and a multilayer perceptron is used to compute the score for an answer (the probability of an answer being the best answer to the question). Despite its simplicity, we achieve state-of-the-art performance on this dataset – 37.17 P@1 and 56.82 MRR. We attribute this improved performance to the use of paragraph vector representations (Le and Mikolov, 2014) instead of averaging over word

Proceedings of NAACL-HLT 2016, pages 1290–1295,
San Diego, California, June 12-17, 2016.

vectors, and to the use of suitable data for training these representations.

2 Approach

2.1 Learning Algorithm

We use a simple feedforward neural network, i.e. a multilayer perceptron, to predict the best answer. As shown in Figure 1, the first layer of the network is a projection layer that transforms question-answer pairs into their vector representations. The vector representation for a question-answer pair (q, a) is a concatenation of the distributed representations q and a for the question and the answer respectively. Each representation is a real-valued vector of a fixed dimensionality d, which is a parameter to be tuned. The projection layer is followed by one or more hidden layers, the number of layers and units in each of these layers are also parameters to be experimentally tuned. We use the rectified linear (ReLU) activation function. Finally, a softmax layer is used to compute the output probability p, i.e. the probabilities p_1 and p_2 of the negative (i.e. not best answer) and positive (i.e. best answer) classes respectively. For each question, all its user-generated answers are ranked according to their probability of being the best answer, as predicted by the network.

Given a question-answer pair (q, a), the possible values for the ground-truth label are 1 (best answer) and 0 (not a best answer). The network is trained by minimizing the L2-regularized cross-entropy loss function between the ground-truth labels and the network predictions on the training set. We use stochastic gradient descent to minimize the loss over the training set. The development set is used for early stopping.

2.2 Document Representations

Our approach requires question-answer pairs to be represented as a fixed-size vector. We experimentally evaluate the Paragraph Vector model (PV) proposed by Le and Mikolov (2014). The PV is an extension of the widely used continuous bag-of-words (CBOW) and skip-gram word embedding models, known as *word2vec*. However, in contrast to CBOW and skip-gram models that only learn word embeddings, the PV is able to learn representations for pieces of text of arbitrary length, e.g. sentences, paragraphs or documents. The PV includes (1) the distributed memory (DM) model, that predicts the next word using the concatenation of the previous words and the paragraph vector, that is shared among all words in the same paragraph (or sentence); (2) the distributed bag-of-words (DBOW) model, that – similar to the skip-gram model – predicts words randomly sampled from the paragraph, given the paragraph vector. We experiment with both DM and DBOW models, as well as their combination. For comparison with recent work in answer reranking (Jansen et al., 2014; Sharp et al., 2015), we also evaluate the averaged word embedding vectors obtained with the skip-gram model (Mikolov et al., 2013) (henceforth referred to as the *SkipAvg* model).

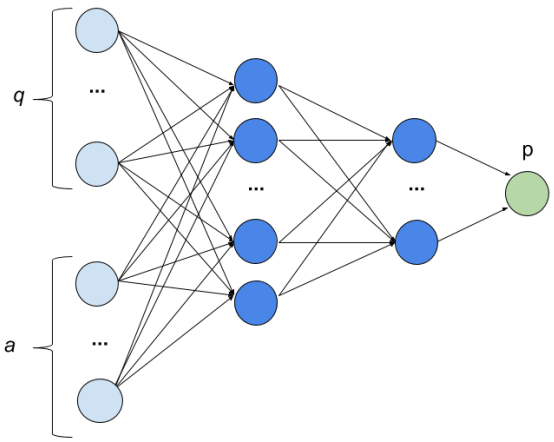

Figure 1: Neural network architecture used to predict answer ranking.

3 Experiments

3.1 Data

In order to be able to compare our work with previous research, we use the Yahoo! Answers dataset that was first introduced by Jansen et al. (2014) and was later used by Sharp et al. (2015) and Fried et al. (2015). This dataset contains 10K *How* questions from Yahoo! Answers. Each question has at least four user-generated answers, and the average number of answers per question is nine. 50% of the dataset is used for training, 25% for development and 25% for testing. Further information about the dataset can be found in Jansen et al. (2014).

Our approach requires unlabelled data for unsupervised pre-training of the word and paragraph vectors. For these purposes we use the *L6 Yahoo! Answers Comprehensive Questions and Answers* corpus obtained via Webscope.[4] This dataset contains about 4.5M questions from Yahoo! Answers along with their user-generated answers, and was provided as training data at the recent TREC LiveQA competition (Agichtein et al., 2015), the goal of which was to answer open-domain questions coming from real users in real time.[5] The Yahoo! Answers manner question dataset prepared by Jansen et al. (2014) and described in the previous paragraph, was initially sampled from this larger dataset. We want to emphasize that the L6 dataset is only used for unsupervised pretraining – no meta-information is used in our experiments.

We also experiment with the English Gigaword corpus,[6] which contains data from several English newswire sources. Jansen et al. (2014) used this corpus to train word embeddings, which were then included as features in their answer reranker.

3.2 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` (Salton, 1991) over lemmas; the candidate answers are ranked according to their cosine similarity to the respective question.

We use the *gensim*[7] implementation of the DBOW and DM paragraph vector models. The word embeddings for the SkipAvg model are obtained with *word2vec*.[8] The data was tokenized with the Stanford tokenizer[9] and then lowercased.

To evaluate our models, we use standard implementations of the P@1 and mean reciprocal rank (MRR) evaluation metrics. To evaluate whether the difference between two models is statistically significant, statistical significance testing is performed using one-tailed bootstrap resampling with 10,000 iterations. Improvements are considered to be statistically significant at the 5% confidence level ($p < 0.05$).

Model	dim	P@1	MRR
Random Baseline	-	15.06	37.13
CR Baseline	-	24.83	48.82
SkipAvg Baseline	200	31.25	52.56
DBOW	100	38.95*	58.18*
DBOW	200	**39.91***	**58.68***
DBOW	300	39.47*	58.35*
DM	100	38.19*	57.01*
DM	200	38.35*	57.28*
DM	300	37.55*	56.67*
DBOW+DM	200	**40.55*#**	**59.12*#**
DBOW+SkipAvg	200	**40.39*#**	**58.91*#**
DBOW+DM+SkipAvg	200	**40.63*#**	**59.14*#**

Table 1: Development P@1 and MRR for different vectors representations. * indicates that improvements over the baselines are statistically significant with $p < 0.05$. # indicates that the improvement over the DBOW model with 200-dimensional vectors is not statistically significant. All significance tests are performed with one-tailed bootstrap resampling with 10,000 iterations.

3.3 Results

In Table 1, we report best development P@1 and MRR of the multilayer perceptron trained on Yahoo! Answers (Jansen et al., 2014) data. Early stopping is used to maximize P@1 on the development set. The distributed representations, including the SkipAvg model, beat both random and candidate retrieval baselines by a large and statistically significant margin. Likewise, the multilayer perceptron with DBOW and DM representations significantly outperform the SkipAvg representations. Both paragraph vector representations initially proposed by Le and Mikolov (2014) – DBOW and DM – provide similarly high performance, however the DBOW model performs slightly better, with the improvement over the DM model being statistically significant. Different dimensionalities of the pretrained vectors provide similar results, with

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

[5] https://sites.google.com/site/trecliveqa2015/

[6] https://catalog.ldc.upenn.edu/LDC2003T05

[7] https://radimrehurek.com/gensim/models/doc2vec.html

[8] https://code.google.com/p/word2vec/

[9] http://nlp.stanford.edu/software/tokenizer.shtml

Model	P@1	MRR
Random Baseline	15.74	37.40
CR Baseline	22.63	47.17
SkipAvg	30.25	51.59
Jansen et al. (2014)	30.49	51.89
Fried et al. (2015)	33.01	53.96
DBOW	37.02*	56.74*
DBOW+DM	37.06*	56.56*
DBOW+SkipAvg	35.85*	56.03*
DBOW+DM+SkipAvg	**37.17***	**56.82***

Table 2: Test P@1 and MRR. * indicates that improvements over the baselines are statistically significant with $p < 0.05$.

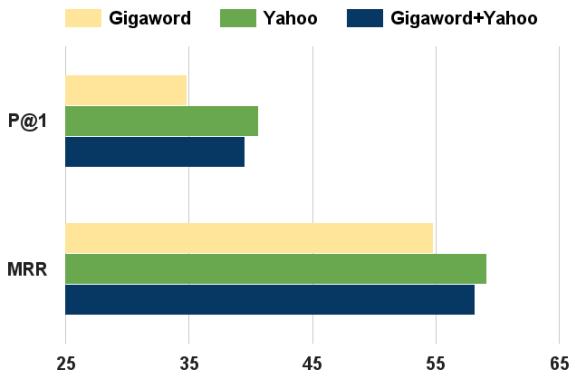

Figure 2: Development P@1 and MRR of a DBOW model pretrained on Yahoo! Answers and Gigaword corpora.

200 outperforming the rest by a small margin. The multilayer perceptron with combinations of different distributed representations reach slightly higher P@1 and MRR on the development set. However, these improvements over the 200-dimension DBOW model are not statistically significant.

Table 2 presents the results on the test set. We only evaluate the 200-dimension DBOW model and its combinations with other models, comparing these to the baselines and the previous results on the same dataset (we use the same train/dev/test split as Jansen et al. (2014)). The DBOW outperforms the baselines by a statistically significant margin. The combination of the DBOW, DM and SkipAvg models provides slightly better results, but the improvement over the DBOW is not statistically significant.

3.4 Analysis

Jansen et al. (2014) report that answer reranking benefits from lexical semantic models, and describe experiments using SkipAvg embeddings pretrained using the English Gigaword corpus. Here we compare the performance of the reranker with distributed representations pretrained on a large "out-of-domain" newswire corpus (Gigaword), versus a smaller "in-domain" non-factoid QA one (L6 Yahoo). Figure 2 shows the development P@1 and MRR of the multilayer perceptron with DBOW model on the Yahoo! Answers dataset pretrained on 30M random paragraphs from the English Gigaword corpus versus the multilayer perceptron with DBOW model pretrained on the Yahoo L6 corpus containing about 8.5M paragraphs. We also evaluate the combination of the two models. The results highlight the importance of finding a suitable source of unlabelled training data since vectors pretrained on reasonably large amounts of Yahoo! Answers data are more beneficial than using a much larger Gigaword dataset.

Even our best model is still, however, far from being perfect, i.e. for about 60% of questions, the answer selected as best by the author of the question is not assigned the highest rank by our system. We believe that one of the reasons for that is that the choice of the best answer purely relies on the question's author and may be subjective (see Table 3). A possible useful direction for future research is to incorporate the user-level information into the neural reranking model. This approach has been recently found beneficial in the task of sentiment analysis (Tang et al., 2015).

Another potential source of error lies in the user-generated nature of the data. Yahoo! Answers contains a large number of spelling and grammar mistakes (e.g. *how do i thaw fozen [sic] chicken?*), non-standard spelling and punctuation (e.g. *Booorrrriinng!!!!!*). A common way to deal with this problem is normalization (Baldwin et al., 2015). To determine whether this might be helpful, we normalized the data following the strategy described by Le Roux et al. (2012). We trained the DBOW model with 200 dimensions and applied the MLP, as described in Section 2. The best development P@1 was only 33.95, with MRR 54.23 (versus 39.91 P@1 and 58.68 MRR without normalization). Even

Question	*How should I wear my hair tomorrow?*
Best answer	*Very good question...Lets see, I think you should wear it in pigtails.....*
Other answers	*Losen it.*
	Close your eyes, grab some scissors, and GO CRAZY!
	I think you should scrunch it! It looks awesome. Just tip ur head over and put jell in ur hands and like scrunch!
	just brush it and go, it always works for me when i can't figure out what to do with it.
	pull it up in a high pony tail & small curls falling down!
	make it into braids

Table 3: Example question from the Yahoo! Answers dataset

though our preliminary experiments show that applying lexical normalization results in significantly lower performance, further study is needed. One direction is in using character-level embeddings that have been proven promising for user-generated content because of their ability to better handle spelling variation (Kim et al., 2015).

4 Conclusions

We have conducted answer reranking experiments for open-domain non-factoid QA and achieved state-of-the-art performance on the Yahoo! Answers manner question corpus using a very straightforward neural approach which involves representing question-answer pairs as paragraph vectors and training a multilayer perceptron to order candidate answers. Our experiments show that representing the question-answer pair as a paragraph vector is clearly superior to the use of averaged word vectors. We have also shown that a smaller amount of unlabelled data taken from a CQA site is more useful for training representations than a larger newswire set.

In this paper, we use general purpose distributed document representations provided by Paragraph Vector models to represent question-answer pairs. Then a machine learning algorithm is used to rank the pairs. One possible direction for future research is in learning distributed document representations and the ranking simultaneously and applying more sophisticated recurrent models such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) neural networks, that have been shown to be effective in similar tasks (Wang and Nyberg, 2015; Zhou et al., 2015).

Acknowledgments

Thanks to Yvette Graham and the three anonymous reviewers for their helpful comments and suggestions. This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of ADAPT centre (`www.adaptcentre.ie`) at Dublin City University. 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

Eugene Agichtein, David Carmel, Dan Pelleg, Yuval Pinter, and Donna Harman. 2015. Overview of the TREC 2015 LiveQA Track. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC*.

Timothy Baldwin, Young-Bum Kim, Marie Catherine de Marneffe, Alan Ritter, Bo Han, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. *ACL-IJCNLP 2015*, page 126.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615, to appear at AAAI 2016*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

Joseph Le Roux, Jennifer Foster, Joachim Wagner, Rasul Samad Zadeh Kaljahi, and Anton Bryl. 2012. DCU-Paris13 systems for the SANCL 2012 shared task. *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR Workshop*.

Gerard Salton. 1991. Developments in automatic text retrieval. *Science*, 253(5023):974–980.

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.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023, Beijing, China, July. Association for Computational Linguistics.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 707–712, Beijing, China, July. Association for Computational Linguistics.

Xiaoqiang Zhou, Baotian Hu, Qingcai Chen, Buzhou Tang, and Xiaolong Wang. 2015. Answer sequence learning with neural networks for answer selection in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 713–718, Beijing, China, July. Association for Computational Linguistics.