Effective Shared Representations with Multitask Learning for Community Question Answering

Daniele Bonadiman[†] and **Antonio Uva**[†] and **Alessandro Moschitti**
[†]DISI, University of Trento, 38123 Povo (TN), Italy
Qatar Computing Research Institute, HBKU, 34110, Doha, Qatar
{d.bonadiman,antonio.uva}@unitn.it
amoschitti@gmail.com

Abstract

An important asset of using Deep Neural Networks (DNNs) for text applications is their ability to automatically engineer features. Unfortunately, DNNs usually require a lot of training data, especially for high-level semantic tasks such as community Question Answering (cQA). In this paper, we tackle the problem of data scarcity by learning the target DNN together with two auxiliary tasks in a multitask learning setting. We exploit the strong semantic connection between selection of comments relevant to (i) new questions and (ii) forum questions. This enables a global representation for comments, new and previous questions. The experiments of our model on a SemEval challenge dataset for cQA show a 20% relative improvement over standard DNNs.

1 Introduction

Deep Neural Networks (DNNs) have successfully been applied for text applications, e.g., (Goldberg, 2015). Their capacity of automatically engineering features is one of the most important reasons for explaining their success in achieving state-of-the-art performance. Unfortunately, they usually require a lot of training data, especially when modeling high-level semantic tasks such as QA (Yu et al., 2014), for which, more traditional methods achieve comparable if not higher accuracy (Tymoshenko et al., 2016a).

Finding a general solution to data scarcity for any task is an open issue, however, for some classes of applications, we can alleviate it by making use of multitask learning (MTL). Recent work has shown that it is possible to *jointly train* a general system for solving different tasks simultaneously. For example, Collobert and Weston (2008) used MTL to train a neural network for carrying out many sequence labeling tasks (e.g., pos-tagging, named entity recognition, etc.), whereas Liu et al. (2015) trained a DNN with MTL to perform multi-domain query classification and reranking of web search results with respect to user queries.

The above work has shown that MTL can be effectively used to improve NNs by leveraging different kinds of data. However, the obtained improvement over the base DNN was limited to 1-2 points, raising the question if this is the kind of enhancement we should expect from MTL. Analyzing the different tasks involved in the model by Liu et al. (2015), it appears evident that query classification provides little and very coarse information to the document ranking task. Indeed, although, the vectors of queries and documents lie in the same space, the query classifier only chooses between four different categories, *restaurant*, *hotel*, *flight* and *nightlife*, whereas the documents can potentially span infinite subtopics.

In this paper, we conjecture that when the tasks involved in MTL are more semantically connected a larger improvement can be obtained. More specifically, MTL can be more effective when we can encode the instances from different tasks using the same representation layer expressing *similar semantics*. To demonstrate our hypothesis, we worked on Community Question Answering (cQA), which is an interesting and relatively new problem and still focused on a query and retrieval setting.

2 Preliminaries and paper results

cQA websites enable users to freely ask questions in web forums and get some good answers in the form of comments from other users. In particu-

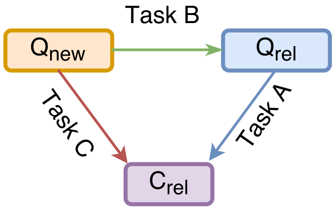

Figure 1: The three tasks of cQA at SemEval: the arrows show the relations between the new and the related questions and the related comments.

lar, given a fresh user question, q_{new}, and a set of forum questions, Q, answered by a comment set, C, the main task consists in determining whether a comment $c \in C$ is a suitable answer to q_{new} or not. Interestingly, the task can be divided into three sub-tasks as shown in Fig. 2: given q_{new}, the main Task C is about directly retrieving a relevant comment from the entire forum data. This can also be achieved by solving Task B to find a similar question, q_{rel}, and then executing Task A to select comments, c_{rel}, relevant to q_{rel}.

Given the above setting, we define an MTL model that solves Task C, learning at the same time the auxiliary tasks A and B. Considering that (i) q_{new} and q_{rel} have the same nature and (ii) comments tend to be short and their text is comparable to the one of questions,[1] we could model an effective shared semantic representation. Indeed, our experiments with the data from SemEval 2016 Task 3 (Nakov et al., 2016) show that our MTL approach improves the single DNN for solving Task C by roughly 8 points in MAP (almost 20% of relative improvement). Finally, given the strong connection between the objective functions of the DNNs, we could train our network with the three different tasks at the same time, performing a single forward-backward operation over the network.

3 Our MTL model for cQA

MTL aims at learning several related tasks at the same time to improve some (or possibly all) tasks using joint information (Caruana, 1997). MTL is particularly well-suited for modeling Task C as it is a composition of tasks A and B, thus, it can benefit from having both questions q_{new} and q_{rel} in input to better model the interaction between the new question and the comment. More precisely, it can use the triplets, $\langle q_{new}, q_{rel}, c_{rel}\rangle$, in the learning process, where the interaction between the triplet members is exploited during the joint training of the three models for the tasks A, B and C. In fact, a better model for question-comment similarity or question-question similarity can lead to a better model for new question-comment similarity (Task C).

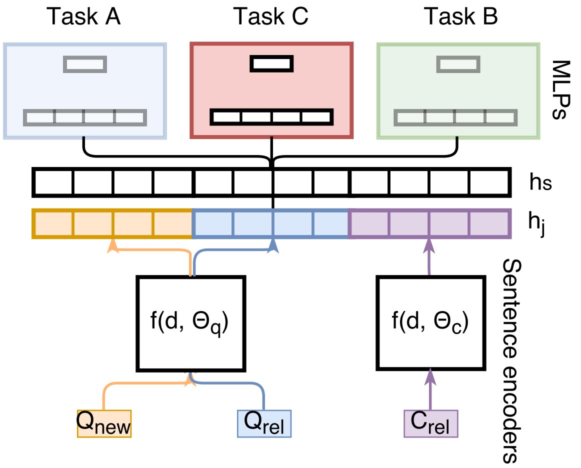

Figure 2: Our MTL architecture for cQA. Given the input sentences q_{new}, q_{rel} and c_{rel} (at the bottom), the NN passes them to the sentence encoders. Their output is concatenated into a new vector, h_j, and fed to a hidden layer, h_s, whose output is passed to three independent multi-layer perceptrons. The latter produce the scores for the individual tasks.

Additionally, each thread in the SemEval dataset is annotated with the labels for all the three tasks and therefore it is possible to apply joint learning directly (using a global loss), rather than training the network by optimizing the loss of the three single tasks independently. Note that, in previous work (Collobert and Weston, 2008; Liu et al., 2015), each input example was annotated for only one task and thus training the model required to alternate examples from the different tasks.

3.1 Joint Learning Architecture

Our joint learning architecture is depicted in Figure 2, it takes three pieces of text as input, i.e, a new question, q_{new}, the related question, q_{rel}, and its comment, c_{rel}, and produces three fixed size representations, $x_{q_{new}}$, $x_{q_{rel}}$ and $x_{c_{rel}}$, respectively. This process is performed using the sentence encoders, $x_d = f(d, \theta_d)$, where d is the input text and θ_d is the set of parameters of the sentence encoder. In previous work, different sentence encoders have been proposed, e.g., Convolutional Neural Networks (CNNs) with max-pooling (Kim, 2014; Severyn and Moschitti, 2015)

[1] In cQA domains, these are typically longer than standard questions, i.e., up to few paragraphs containing subquestions and an introduction.

	Task A	Task B	Task C
Train	37.51%	39.41%	9.9%
Train + ED	37.47%	64.38%	21.25%
Dev	33.52%	42.8%	6.9%
Test	40.64%	33.28%	9.3%

Table 1: Percentage of positive examples in the training datasets for each task.

and Long-short term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997).

We concatenate the three representations, $h_j = [x_{q_{new}}, x_{q_{rel}}, x_{c_{rel}}]$, and fed them to a hidden layer to create a shared input representation for the three tasks, $h_s = \sigma(Wh_j + b)$. Next, we connect the output of h_s to three independent Multi-Layer Perceptrons (MLP), which produce the scores for the three tasks. At training time, we compute the global loss as the sum of the individual losses for the three tasks for each example, where each loss is computed as binary cross-entropy.

3.2 Shared Sentence Models

The SemEval dataset contains ten times less new questions than related questions by construction. However, all questions have the same nature (i.e., generated by forum users), thus, we can share the parameters of their sentence models as depicted in Figure 2. Formally, let $x_d = f(d, \theta)$ be a sentence model for a text, d, with parameters, θ, i.e., the embedding weights and the convolutional filters: in a standard setting, each sentence model uses a different set of parameters $\theta_{q_{new}}$, $\theta_{q_{rel}}$ and $\theta_{c_{rel}}$. In contrast, our proposed sentence model encodes both the questions, q_{new} and q_{rel}, using the same set of parameters θ_q.

4 Experiments

4.1 Setup

Dataset: the data for the above-mentioned tasks is distributed in three datasets for: Task A, which contains $6,938$ related questions and $40,288$ comments. Each comment in the dataset was annotated with a label indicating its relevancy to the question of its thread. Task B, which contains 317 new questions. For each new question, 10 related questions were retrieved, summing to $3,169$ related questions. Also in this case, the related questions were annotated with a relevancy label, which tells if they are relevant to the new question or not. Task C contains 317 new questions, together with $3,169$ related questions (same as in Task B) and $31,690$ comments. Each comment was labeled with its relevancy with respect to the new question. Each of the three datasets is in turn divided in training, dev. and test sets.

Model	MAP	MRR
LSTM	43.91	49.28
CNN	44.43	49.01
CNN Train	44.43	49.01
CNN Train + ED[3]	**44.77**	**52.07**

Table 2: Impact of CNN vs. LSTM sentence models on the baseline network for Task C.

Table 1 reports the label distributions with respect to the different datasets. The data for Task C presents a higher number of negative than positive examples. Thus, we automatically extended the set of positive examples in our joint MTL training set using the data from Task A. More specifically, we take the pair (q_{rel}, c_{rel}) from the training set of Task A and create the triples, $(q_{rel}, q_{rel}, c_{rel})$, where the label for question-question similarity is obviously positive and the labels for Task C are inherited from those of Task A. We ensured that the questions in the extended data (ED) generated from the training set do not overlap with questions from the dev. and test sets. The resulting training data contains $34,100$ triples: its relevance label distribution is shown in the row, Train + ED, of Table 1. [2]

Pre-processing: we tokenized and put both questions and comments in lowercase. Moreover, we concatenated question subject and body to create a unique question text. For computational reasons, we limited the document size to 100 words. This did not cause any degradation in accuracy.

Neural Networks: we mapped words to embeddings of size 50, pre-initializing them with standard skipgram embeddings of dimensionality 50. The latter embeddings were trained on the English Wikipedia dump using word2vec toolkit (Mikolov et al., 2013). We encoded the input sentence with a fixed-sized vector, whose dimensions are 100, using a convolutional operation of size 5 and a k-max pooling operation with $k = 1$. Table 2 shows the results of our preliminary experiments with the sentence models of CNN and LSTM, respectively, on the dev. set of Task C. In our further experiments, we opted for CNN since it produced a bet-

[2] We make out MTL data available at `http://ikernels-portal.disi.unitn.it/repository/`

[3] Extended Dataset for Task C computed using the questions from Task A.

Model	DEV		TEST	
	MAP	MRR	MAP	MRR
Random	-	-	15.01	15.19
IR Baseline	-	-	40.36	45.83
SUper-team	-	-	55.41	61.48
KeLP	-	-	52.95	59.23
SemanticZ	-	-	51.68	55.96
MTE-NN	-	-	49.38	51.56
ICL00	-	-	49.19	53.89
SLS	-	-	49.09	55.98
ITNLP-AiKF	-	-	48.49	55.21
ConvKN	-	-	47.15	51.43
ECNU	-	-	46.47	51.41
UH-PRHLT	-	-	43.20	47.79
$\langle q_{new}, c_{rel} \rangle$	44.77	52.07	41.95	47.21
$\langle q_{new}, q_{rel}, c_{rel} \rangle$	45.59	51.04	46.99	55.64
$\langle q_{new}, q_{rel}, c_{rel} \rangle$ + $\leftrightarrow$	47.82	53.03	46.45	51.72
MTL (BC)	47.80	52.31	48.58	55.77
MTL (AC)	46.34	51.54	48.49	54.01
MTL (ABC)	**49.63**	55.47	**49.87**	55.73
MTL + one feature	-	-	**52.67**	55.68

Table 3: Results on the validation and test sets for the proposed models.

ter MAP and is computationally more efficient.

For each MLP, we used a non-linear hidden layer (with hyperbolic tangent activation, Tanh), whose size is equal to the size of the previous layer, i.e., 100. We included information such as word overlaps (Tymoshenko et al., 2016a) and rank position as embeddings with an additional lookup table with vectors of size $d_{feat} = 5$. The rank feature is provided in the SemEval dataset and describes the position of the questions/comments in the search engine output.

Training: we trained our networks using SGD with shuffled mini-batches using the rmsprop update rule (Tieleman and Hinton, 2012). We set the training to iterate until the validation loss stops improving, with patience $p = 10$, i.e., the number of epochs to wait before early stopping, if no progress on the validation set is obtained. We added dropout (Srivastava et al., 2014) between all the layers of the network to improve generalization and avoid co-adaptation of features. We tested different dropout rates (0.2, 0.4) for the input and (0.3, 0.5, 0.7) the hidden layers obtaining better results with highest values, i.e., 0.4 and 0.7.

4.2 Results

Table 3 shows the results of our individual and MTL models, in comparison with the Random and IR baselines of the challenge (first two rows), and the SemEval 2016 systems (rows 3–12). Rows 13-15 illustrate the results of our models when trained only on Task C. $\langle q_{new}, c_{rel} \rangle$ corresponds to the basic model, i.e., the single network, whereas the $\langle q_{new}, q_{rel}, c_{rel} \rangle$ model only exploits the joint input, i.e., the availability of q_{rel}. Rows 16-18 report the MTL models combining Task C with the other two tasks. The difference with the previous group (rows 13-15) is in the training phase, which is also operated on the instances from tasks A and B.

We note that: (i) the single network for Task C cannot compete with the challenge systems, as it would be ranked at the last position, according to the official MAP score (test set result); (ii) the joint representation, $\langle q_{new}, q_{rel}, c_{rel} \rangle$, highly improves the MAP of the basic network from 41.95 to 46.99 on the test set. This confirms the importance of having highly related tasks using input encoding closely related semantics. (iii) The shared sentence model for q_{new} and q_{rel} (indicated with $\leftrightarrow$) improves MAP on the dev. set only. (iv) The MTL (ABC) provides the best MAP, improving BC and AC by 1.29 and 1.38, respectively. Most importantly, it also improves, $\langle q_{new}, q_{rel}, c_{rel} \rangle$ by 2.88 points, i.e., the best model using the joint representation and no training on the auxiliary tasks.

Additionally, our full MTL model would have ranked 4^{th} on Task C of the SemEval 2016 competition. This is an important result since all the challenge systems make use of many manually engineered features whereas our model does not (except for the necessary initial rank). If we add the most powerful feature used by the top systems to our model, i.e., the weighted sum between the score of the Task A classifier and the Google rank (Mihaylova et al., 2016; Filice et al., 2016), our system would achieve an MAP of 52.67, i.e., very close to the second system.

Finally, we do not report the results of the auxiliary tasks for lack of space and also because our idea of using MTL is to improve the target Task C. Indeed, by their definition, tasks A and B are simpler than C, and are designed for solving it. Thus, attempting to improve the simpler A and B tasks by solving the more complex Task C, although interesting, looks less realistic. Indeed, we did not observe any important improvement of tasks A and B in our MTL setting. More insights and results are available in our longer version of this paper (Bonadiman et al., 2017).

5 Related Work

The work related to cQA spans two major areas: question and answer passage retrieval. Hereafter, we report some important research about them and then conclude with specific work on MTL.

Question–Question Similarity. Early work on question similarity used statistical machine translation techniques, e.g., (Jeon et al., 2005; Zhou et al., 2011), to measure similarity between questions. Language models for question-question similarity were explored by Cao et al. (2009), who incorporated information from the category structure of Yahoo! Answers when computing similarity between two questions. Instead, Duan et al. (2008) proposed an approach that identifies the topic and focus from questions and compute their similarity. Ji et al. (2012) and Zhang et al. (2014) learned a probability distribution over the topics that generate the question/answers pairs with LDA and used it to measure similarity between questions. Recently, Da San Martino et al. (2016) showed that combining tree kernels (TKs) with text similarity features can improve the results over strong baselines such as Google.

Question–Answer Similarity. Yao et al. (2013) used a conditional random field trained on a set of powerful features, such as tree-edit distance between question and answer trees. Heilman and Smith (2010) used a linear classifier exploiting syntactic features to solve different tasks such as recognizing textual entailment, paraphrases and answer selection. Wang et al. (2007) proposed Quasi-synchronous grammars to select short answers for TREC questions. Wang and Manning (2010) used a probabilistic Tree-Edit model with structured latent variables for solving textual entailment and question answering. Severyn and Moschitti (2012) proposed SVM with TKs to learn structural patterns between questions and answers encoded in the form of shallow syntactic parse trees, whereas in (Tymoshenko et al., 2016b; Barrón-Cedeño et al., 2016) the authors used TKs and CNNs to rank comments in web forums, achieving the state of the art on the SemEval cQA challenge. Wang and Nyberg (2015) trained a long short-term memory model for selecting answers to TREC questions.

Finally, a recent work close to ours is (Guzmán et al., 2016), which builds a neural network for solving Task A of SemEval. However, this does not approach the problem as MTL.

Related work on MTL. A good overview on MTL, i.e., learning to solve multiple tasks by using a shared representation with mutual benefit, is given in (Caruana, 1997). Collobert and Weston (2008) trained a convolutional NN with MTL which, given an input sentence, could perform many sequence labeling tasks. They showed that jointly training their system on different tasks, such as speech tagging, named entity recognition, etc., significantly improves the performance on the main task, i.e., semantic role labeling, without requiring hand-engineered features.

Liu et al. (2015) is the closest work to ours. They used multi-task deep neural networks to map queries and documents into a semantic vector representation. The latter is later used into two tasks: query classification and question-answer reranking. Their results showed a competitive gain over strong baselines. In contrast, we have presented a model that can also exploit a joint question and comment representation as well as the dependencies among the different SemEval Tasks.

6 Conclusions

We proposed an MTL architecture for cQA, where we could exploit auxiliary tasks, which are highly semantically connected with our main task. This enabled the use of the same semantic representation for encoding the text objects associated with all the three tasks, i.e., new question, related question and comments. Our shared semantic representation provides an important advantage over previous MTL applications, whose subtasks share a less consistent semantic representation.

Our experiments on the SemEval 2016 dataset show that our MTL approach relatively improves the individual DNNs by almost 20%. This is due to the shared representation as well as training on the instances of the two auxiliary tasks.

In the future, we would like to experiment with hierarchical MTL for stressing even more the role of the auxiliary tasks with respect to the main task. Additionally, we would like to apply constraints on the global loss for enforcing specific relations between the tasks.

Acknowledgements

This work has been partially supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action). Many thanks to the anonymous reviewers for their valuable suggestions.

References

Alberto Barrón-Cedeño, Daniele Bonadiman, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad A 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. *Proceedings of SemEval*, pages 896–903.

Daniele Bonadiman, Antonio Uva, and Alessandro Moschitti. 2017. Multitask Learning with Deep Neural Networks for Community Question Answering. *ArXiv e-prints*, February.

Xin Cao, Gao Cong, Bin Cui, Christian Søndergaard Jensen, and Ce Zhang. 2009. The use of categorization information in language models for question retrieval. In *CIKM*.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Giovanni Da San Martino, Alberto Barrón Cedeño, Salvatore Romeo, Antonio Uva, and Alessandro Moschitti. 2016. Learning to re-rank questions in community question answering using advanced features. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1997–2000. ACM.

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *ACL*.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. *Proceedings of SemEval*, 16:1116–1123.

Yoav Goldberg. 2015. A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726.

Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2016. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 460–466, Berlin, Germany, August. Association for Computational Linguistics.

Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics.

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

Jiwoon Jeon, W Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *CIKM*.

Zongcheng Ji, Fei Xu, Bin Wang, and Ben He. 2012. Question-answer topic model for question retrieval in community question answering. In *CIKM*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, Doha, Qatar, October.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proc. NAACL*.

Tsvetomila Mihaylova, Pepa Gencheva, Martin Boyanov, Ivana Yovcheva, Todor Mihaylov, Momchil Hardalov, Yasen Kiprov, Daniel Balchev, Ivan Koychev, Preslav Nakov, Ivelina Nikolova, and Galia Angelova. 2016. Super team at semeval-2016 task 3: Building a feature-rich system for community question answering. In *SemEval@NAACL-HLT*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

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. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 741–750. ACM.

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. ACM.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

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

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2016b. Learning to rank non-factoid answers: Comment selection in web forums. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2049–2052. ACM.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1164–1172, Beijing, China.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. *ACL, July*.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, pages 858–867. Citeseer.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *CoRR*.

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *CIKM*.

Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *ACL*.