

Semantic Linking in Convolutional Neural Networks for Answer Sentence Selection

Massimo Nicosia* and Alessandro Moschitti†

*DISI, University of Trento, 38123 Povo (TN), Italy

Amazon, Manhattan Beach, CA, USA, 90266

m.nicosia@gmail.com amosch@amazon.com

Abstract

State-of-the-art networks that model relations between two pieces of text often use complex architectures and attention. In this paper, instead of focusing on architecture engineering, we take advantage of small amounts of labelled data that model semantic phenomena in text to encode matching features directly in the word representations. This greatly boosts the accuracy of our reference network, while keeping the model simple and fast to train. Our approach also beats a tree kernel model that uses similar input encodings, and neural models which use advanced attention and compare-aggregate mechanisms.

1 Introduction

Modeling a match between pieces of text is at the core of many NLP tasks. Recently, manual feature engineering methods have been shadowed by neural network approaches. These networks model the interaction of two pieces of text, or word-to-word interactions across sentences, using sophisticated attention mechanisms (Wang et al., 2016a; Santos et al., 2016) and compare-aggregate frameworks (He and Lin, 2016; Wang et al., 2017).

Architectural complexity is tied to longer training times¹. Meaningful features may take long time to emerge by only leveraging word representations and the training data of the task at hand. This is especially problematic with little data, as it often happens in question answering (QA) tasks, e.g., answer sentence selection (Wang et al., 2007; Yang et al., 2015). Thus, effective word representations are crucial in neural network models to get state-of-the-art performance.

*Now at Google

†This work was partially carried out when the author was at the University of Trento

¹<http://dawn.cs.stanford.edu/benchmark/>

In this work, we try to answer the following research questions: (i) in addition to lexical links, can we incorporate higher-level semantic links between the words in a question and a candidate answer passage, and (ii) can we show that such information has an impact on the quality of our model, and also allows us to keep the architecture simple?

We show that modeling semantic relations improves the performance of a neural network for answer sentence selection with (i) a little number of semantic annotations, and (ii) a little increase in training time w.r.t. more complex architecture.

2 Related Work

Traditional work on QA makes heavily use of syntactic and semantic features (Hickl et al., 2007; Ferrucci et al., 2010). A different direction consists in using structural kernels on text encoded as trees (Severyn and Moschitti, 2012; Severyn et al., 2013a,b; Tymoshenko et al., 2014; Tymoshenko and Moschitti, 2015). Recently, deep learning methods have been very successful in NLP tasks. Words and sentences are mapped into low dimensional representations using convolutional (Krizhevsky et al., 2012) and recurrent networks (Schuster and Paliwal, 1997), and then adoperated for classification. Complex networks for such a task include attentive networks and compare-aggregate networks.

Attentive Networks (Bahdanau et al., 2015; Parikh et al., 2016; Yin et al., 2016) build a sentence representation by also considering the other sentence, weighting the contribution of its parts with the so-called attention mechanism.

Compare-Aggregate Networks (Wang and Jiang, 2017) apply several decompositions to each sentence in a pair. The resulting vectors are compared or composed with multiple functions, and possibly some attention mechanisms. All the

intermediate results are then aggregated into a fixed size vector to quantify the final match.

In this work, we take some elements of the traditional QA research, i.e., semantic features, and use them to model relationships between sentence pairs, in the context of a neural network, which is less complex than attentive and compare-aggregate counterparts.

3 Question Analysis

Question Analysis is an important part of a QA system (Lally et al., 2012) and can give us syntactic and semantic clues that greatly help in scoring answer passages, and in identifying the final answer. Leveraging a relatively small number of annotated examples, we can automatically extract question properties that may be exploited by a QA model to increase the accuracy of its answers. We use classifiers to extract the *question category* and the *question focus*.

Question Category. Questions can be broadly classified into categories according to a given taxonomy. When the category is indicative of the answer type, the latter can be furtherly characterized by the Lexical Answer Type (LAT), which according to Lally et al. (2012) is a word or noun phrase in the question that specifies the type of the answer without any attempt to understand its semantics.

Question Focus. In the literature there are multiple definitions of question focus. According to Ferrucci et al. (2010), the focus is the question part that substituted with the answer, renders the question a stand-alone statement. According to Bunescu and Huang (2010), the focus is the “set of all maximal noun phrases in the question that corefer with the answer”. Their definition allows a question to have multiple focuses or an implicit focus. Additionally, it is more tied to the LAT and indeed the focus can be used to infer the answer type. We adopt such definition since we build our question focus identifier using the annotated data they provide. Note that we do not consider multiword or implicit focus.

4 Answer Sentence Selection with CNNs

Given a query or question q and a candidate answer passage a , the task of answer selection can be defined as learning a function $f(q, a)$ that outputs a relevancy probability $s \in [0, 1]$. Multiple answers associated with a question are sorted in descending order by the score s . A good an-

swer selection system places the highest number of correct answers at the top of a candidate answer list. In this paper, we use convolutional neural networks, referred to as CNNs (Kim, 2014; Kalchbrenner et al., 2014), to (i) classify a question into a category, (ii) identify the focus word in a question, and (iii) build a question and answer representations for QA.

4.1 Sentence Matrix Encoding

A sentence s of length n is a sequence of words (w_1, \dots, w_n) , which are drawn from a vocabulary V . Each word is encoded with an integer id from 1 to $|V|$, and then represented as a vector, $\mathbf{w} \in \mathbb{R}^d$, looked up into an embedding matrix, $\mathbf{E} \in \mathbb{R}^{d \times |V|}$. The matrix \mathbf{E} is obtained by concatenating all the embeddings of the words in V . The id 0 is used for padding and it is mapped to the zero vector. The i^{th} column in \mathbf{E} corresponds to the word with integer id i to facilitate the lookup.

4.2 Question Analysis Networks

We use CNNs for question analysis. The **question category network** applies convolutions of different width and then pooling on the question. The results are concatenated and fed to a multi-layer perceptron (MLP) that outputs a probability distribution over the possible categories seen during training. The **question focus network** applies convolutions that operate on windows centered on each question word. Therefore, the input and output resolutions are the same. We stack a number of convolutions to increase the receptive field. Every output vector from the last convolution of the stack is passed through an MLP, which produces a scalar value. All those values are normalized across each sentence with a softmax, to form a probability distribution over the sentence tokens.

4.3 Answer Sentence Selection Network

Our neural model is based on the Severyn and Moschitti (2015, 2016) model (S&M from now on), showed in Figure 1. This model is simple, fast and well studied. It has also been reproduced in other work (Rao et al., 2017; Chen et al., 2017; Sequiera et al., 2017).

The S&M model embeds the question and answer passage and operates independent convolutional and max-pooling layers on each. A bilinear transformation (Bordes et al., 2014) produces a similarity value x_{sim} for the pair. The similarity, the encoded question and passage, and a vec-

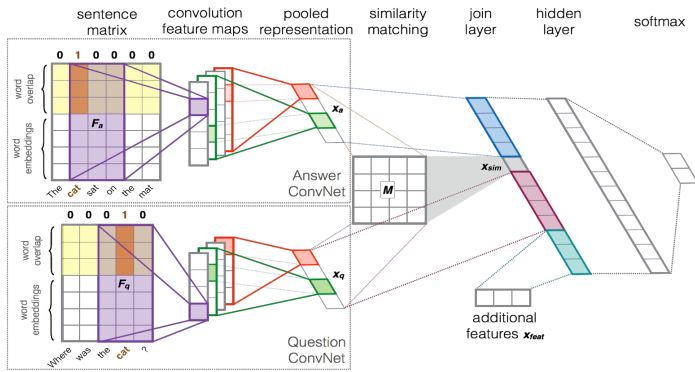


Figure 1: The S&M CNN model. Diagram from [Severyn and Moschitti \(2016\)](#).

tor of real valued features x_{feat} are concatenated in the join layer. The latter is fed to a hidden layer with a non-linearity, and the final softmax layer outputs the matching probability. The word vectors of the question and the answer are augmented with an additional feature, which is embedded in a small dimensional space. This feature signals if a word appears in both the question and answer. We found that the real valued features and the similarity matrix do not increase the network accuracy and we removed them from our model. This finding is consistent with recent reproduction papers by [Rao et al. \(2017\)](#); [Sequiera et al. \(2017\)](#).

4.4 Our QA Network with Semantic Overlap

We propose to add semantic features to the sentence matrix to establish links between words that go beyond lexical matching. Figure 2 describes our network. The key addition to the S&M model is the semantic overlap vector. Each word is therefore represented by concatenating three vectors: the word embedding vector, a feature embedding vector which can represent two values – if a word is contained or not in both question and answer – and the semantic overlap embedding vector. The semantic vector \mathbf{w}_{so} , with dimensionality s , embeds a feature so which can assume $C + 1$ values, if we consider the C question classes plus a no-match value. Each feature value is looked up into an embedding matrix $\mathbf{W}_{so} \in \mathbb{R}^{s \times |C| + 1}$. Analogously, the word overlap binary feature is looked up into an embedding matrix $\mathbf{W}_{wo} \in \mathbb{R}^{r \times |2|}$. The final word representation will be the concatenation of all these vectors: $\mathbf{w}' = [\mathbf{w}; \mathbf{w}_{wo}; \mathbf{w}_{so}]$.

Here we describe how the semantic word overlap feature is computed. For each question we collect the output of our question analysis CNNs. The question focus CNN determines which word in the

Cat.	Named Entity Type
HUM	Person
LOC	Loc, Gpe
NUM	Date, Time, Percent, Quantity, Ordinal, Cardinal
ENTY	Norp, Org, Facility, Product, Event, Work of art, Law, Language
DESC	Norp, Org, Facility, Product, Event, Work of art, Law, Language, Date, Time, Percent, Quantity, Ordinal, Cardinal
ABBR	Norp, Org, Facility, Product, Event, Work of art, Law, Language

Table 1: Mapping between question categories and OntoNotes entity types.

question is the focus. The question category CNN assigns a class to the question. After that, each word is associated with a semantic overlap feature so (which will eventually be embedded using \mathbf{W}_{wo}) according to the following strategy:

1. for each word in the question which is not the question focus so is equal to 0. For the question focus word so is equal to the id of the question category (the question focus and category are output by our CNN classifiers);
2. for each answer word so is equal to 0, with the exception of words covered by named entities (NEs), for which so is equal to the id of the question category that is compatible with their entity type, according to the mapping in Table 1.

The \mathbf{W}_{wo} and \mathbf{W}_{so} matrices are parameters of the model, and they are learned during training. The question category and question focus annotations for the QA datasets are produced by our neural network classifiers. The NEs are obtained with an off-the-shelf processor², trained on OntoNotes ([Weischedel et al., 2012](#)).

5 Experimental Results

Here we describe how we train our networks for question analysis and then we present the answer sentence selection experiments. More details about preprocessing, training and hyperparameter choice can be found in the appendix.

5.1 Question Classification

Dataset. The CNN question classifier is trained on the UIUC dataset ([Li and Roth, 2006](#)). We use the 6 coarse classes to train the classifier.

²In the future, we will also train the NE recognizer.

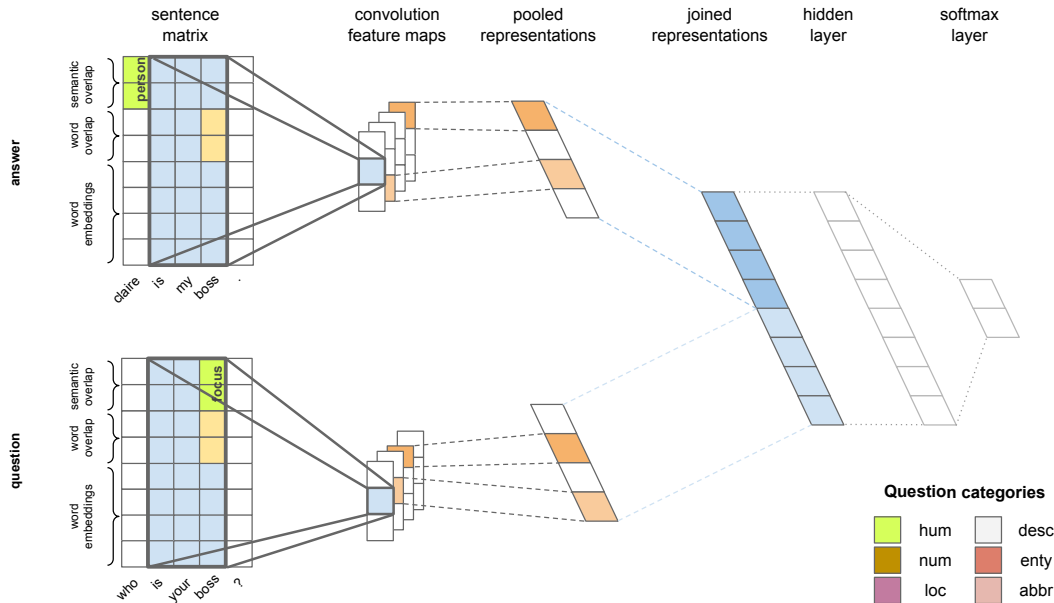


Figure 2: Our model with word and semantic overlap vectors. The convolution of size 3 is not padded, and the filters are 4. The semantic overlap vectors of the question focus word *boss*, and the answer word *claire* are the same, because the latter is an entity of type *Person*. The question has HUM category. Ignoring stopwords, the word *boss* appears in the question and the answer, and this is reflected in the word overlap embedding space.

Results. The classifier has accuracy of 91.2% on the UIUC test set. Our goal is to annotate new questions with reasonable accuracy. Since the model converges well, we annotate the questions in the QA datasets after training on the UIUC data, and select the best model on the test data.

5.2 Question Focus Identification

Dataset. The CNN focus identifier is trained on the dataset from [Bunescu and Huang \(2010\)](#), which contains the first 2,000 UIUC questions annotated with focus information. After removing the questions with implicit and multi-focus, we end up with 1,030 questions.

Results. The cross-validation accuracy of the classifier is 92.3%. After convergence, we annotate the focus words in the QA datasets.

5.3 TrecQA

Dataset. We test our model on TrecQA ([Wang et al., 2007](#)), one of the most popular benchmarks for answer selection. The dataset contains factoid questions and candidate answer sentences.

We use the same splits of the original data, but we run our experiments using the larger provided training set (TRAIN-ALL). This is noisier data, which, on the other hand, gives us more examples for training. We remove from the dev. and test sets questions without answers, and questions with

System	MAP	MRR
Santos et al. (2016)	75.30	85.11
He and Lin (2016)	75.88	82.19
Severyn and Moschitti (2016)	76.54	81.86
Wang et al. (2016b)	77.14	84.47
Rao et al. (2016)	80.10	87.70
Wang et al. (2017)	80.20	87.50
Shen et al. (2017)	82.20	88.90
CNN _{WO} TRAIN-ALL	76.49 (0.4)	84.22 (0.5)
CNN _{WO+SO} TRAIN-ALL	77.93 (0.7)	84.89 (0.9)

Table 2: MAP and MRR (%) on the TrecQA Clean dataset.

only correct or incorrect answer sentence candidates. The resulting dev. and test sets contain respectively 65 and 68 questions. This setting follows a standard in recent work on TREC-QA which is referred to as TrecQA Clean (See [Shen et al. \(2017\)](#)).

Results. Table 2 contains results from previous work, and the performance of our models. CNN_{WO} is our variant of the S&M model. It has comparable performance in terms of MAP, but it is 2.4% points higher in MRR. Our model CNN_{WO+SO} that integrates the semantic overlap improves over CNN_{WO} by 1.44% points in MAP, and 0.67% points in MRR. It approaches the model by [Rao et al. \(2016\)](#), which uses a triplet ranking loss, and several strategies to build

System	MAP	MRR
Miao et al. (2016)	68.86	70.69
Yin et al. (2016)	69.21	71.08
Severyn and Moschitti (2016)	69.51	71.07
Chen et al. (2017)	70.10	71.80
Rao et al. (2016)	70.90	72.30
Tymoshenko et al. (2016)	71.25	72.30
Guo et al. (2017)	71.71	73.36
Wang et al. (2017)	71.80	73.10
Shen et al. (2017)	73.30	75.00
Wang et al. (2016a)	73.41	74.18
Wang and Jiang (2017)	74.33	75.40
CNN _{WO}	69.53 (0.5)	71.35 (0.5)
CNN _{WO+SO}	72.24 (0.5)	73.91 (0.5)

Table 3: MAP and MRR (%) on the WikiQA dataset.

training instances with difficult negative examples. Our system beats several others that use word alignments and attention mechanisms. The better systems employ expensive bidirectional networks, sophisticated attention mechanisms, and extract multiple views of questions and answers for comparing and aggregating them.

5.4 WikiQA

Dataset. TrecQA and its test set are small, so results may be unstable. In addition, lexical overlap between questions and answer candidates is high (Yih et al., 2013). This means that simple lexical similarity features are highly discriminative. Therefore, we also experiment with WikiQA (Yang et al., 2015), which is an order of magnitude larger than TrecQA. We use the Yin et al. (2016) experimental setting.

Results. Table 3 contains the results on WikiQA. Again the MAP score is comparable with the S&M model, while the MRR is slightly higher. Our model CNN_{WO+SO} improves over CNN_{WO} by 2.71% points in MAP, and 2.56% points in MRR, with a higher margin with respect to TrecQA. Interestingly, our approach improves upon the Tymoshenko et al. (2016) tree-kernel model by 1 MAP point. This model includes relational information in terms of question focus, entities and question categories too, but uses additional syntactic information (i.e., syntactic trees). Our network is able to make better use of the provided semantic clues. Surprisingly, CNN_{WO+SO} also achieves higher MAP than the model by Wang et al. (2017), which is a state-of-the-art complex approach mixing attention and interaction factors

of multiple sentence perspectives.

5.5 Discussion

The results with the CNN_{WO+SO} model suggest that the semantic overlap vectors are an effective way of linking questions and answers. This is especially true, given the results on WikiQA, where the questions and answers have little lexical overlap. With the additional semantic information, the CNN is able to better model the relevancy of candidate passages. It also surpasses the accuracy of more complex systems, which have higher training time. The annotation networks (which can be trained only once) and the answer selection networks take little time to train: from 10 to 20 minutes in total, depending on the number of question/answer pairs. CNNs are faster at training and inference time with respect to RNNs, especially when the latter incorporate attention mechanisms, which increase the number of computations. We argue that annotating a relatively small number of examples with semantic information, could be time well spent to increase model accuracy, without increasing its architectural complexity. We would like to add that we also experimented with RNNs (LSTM and GRU) in place of the CNN sentence model. Such encoders easily overfitted, requiring careful regularization, and did not yield better results for us.

6 Conclusion and Future Work

In this paper, we presented a neural network that models semantic links between questions and answers, in addition to lexical links. The annotations for establishing such links are produced by a set of fast neural components for question analysis, trained on publicly available datasets. The evaluation on two QA datasets shows that our approach can achieve state-of-the-art performance using a simple CNN, leading to a low complexity and training time. Our approach is an interesting first step towards a future architecture, in which we will jointly optimize the semantic annotators and the answer sentence selection model, in an end-to-end fashion.

Acknowledgments

The first author was supported by the Google Europe Doctoral Fellowship Award 2015. Many thanks to the anonymous reviewers, the Area and PC Chairs for their valuable work.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR*.
- Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 8724, ECML PKDD 2014*, pages 165–180, New York, NY, USA. Springer-Verlag New York, Inc.
- Razvan Bunescu and Yunfeng Huang. 2010. Towards a general model of answer typing: Question focus identification. In *CICLing*.
- Ruey-Cheng Chen, Evi Yulianti, Mark Sanderson, and W. Bruce Croft. 2017. On the benefit of incorporating external features in a neural architecture for answer sentence selection. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17*, pages 1017–1020, New York, NY, USA. ACM.
- David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.
- Jiahui Guo, Bin Yue, Guandong Xu, Zhenglu Yang, and Jin-Mao Wei. 2017. An enhanced convolutional neural network model for answer selection. In *Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion*, pages 789–790, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.
- Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, San Diego, California. Association for Computational Linguistics.
- Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Ying Shi, and Bryan Rink. 2007. Question answering with lcc’s chaucer at trec 2006. In *Proceedings of the Text REtrieval Conference*, pages 283–292.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.
- A. Lally, JM Prager, MC McCord, BK Boguraev, S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll. 2012. Question analysis: How watson reads a clue. *IBM Journal of Research and Development*, 56(3.4).
- Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249.
- Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736.
- Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proc. of EMNLP*.
- Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM '16*, pages 1913–1916, New York, NY, USA. ACM.
- Jinfeng Rao, Hua He, and Jimmy Lin. 2017. Experiments with convolutional neural network models for answer selection. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1217–1220. ACM.
- Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11).
- Royal Sequiera, Gaurav Baruah, Zhucheng Tu, Salman Mohammed, Jinfeng Rao, Haotian Zhang, and Jimmy J. Lin. 2017. Exploring the effectiveness of convolutional neural networks for answer selection in end-to-end question answering. In *SIGIR 2017 Workshop on Neural Information Retrieval (Neu-IR'17)*.

- 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*, SIGIR '12, pages 741–750, New York, NY, USA. 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.
- Aliaksei Severyn and Alessandro Moschitti. 2016. Modeling relational information in question-answer pairs with convolutional neural networks. *CoRR*, abs/1604.01178.
- Aliaksei Severyn, Massimo Nicosia, and Alessandro Moschitti. 2013a. Building structures from classifiers for passage reranking. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, CIKM '13, pages 969–978, New York, NY, USA. ACM.
- Aliaksei Severyn, Massimo Nicosia, and Alessandro Moschitti. 2013b. Learning adaptable patterns for passage reranking. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 75–83, Sofia, Bulgaria. Association for Computational Linguistics.
- Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.
- 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. Association for Computational Linguistics.
- Kateryna Tymoshenko and Alessandro Moschitti. 2015. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1451–1460, New York, NY, USA. ACM.
- Kateryna Tymoshenko, Alessandro Moschitti, and Aliaksei Severyn. 2014. Encoding semantic resources in syntactic structures for passage reranking. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–672, Gothenburg, Sweden. Association for Computational Linguistics.
- Bingning Wang, Kang Liu, and Jun Zhao. 2016a. Inner attention based recurrent neural networks for answer selection. In *ACL*.
- Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.
- Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *Proc. of ICLR*.
- Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of IJCAI*.
- Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016b. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349. The COLING 2016 Organizing Committee.
- Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, et al. 2012. Ontonotes release 4.0.
- Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.
- Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1744–1753, Sofia, Bulgaria. Association for Computational Linguistics.
- Wenpeng Yin, Hinrich Schutze, Bing Xiang, and Bowen Zhou. 2016. Abcnm: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272.