Automatically Extracting High-Quality Negative Examples for Answer Selection in Question Answering

Haotian Zhang[1], Jinfeng Rao[2], Jimmy Lin[1], and Mark D. Smucker[3]

[1] David R. Cheriton School of Computer Science, University of Waterloo

[2] Department of Computer Science, University of Maryland

[3] Department of Management Sciences, University of Waterloo

{haotian.zhang,jimmylin,mark.smucker}@uwaterloo.ca,jinfeng@cs.umd.edu

ABSTRACT

We propose a heuristic called "one answer per document" for automatically extracting high-quality negative examples for answer selection in question answering. Starting with a collection of question–answer pairs from the popular TrecQA dataset, we identify the original documents from which the answers were drawn. Sentences from these source documents that contain query terms (aside from the answers) are selected as negative examples. Training on the original data plus these negative examples yields improvements in effectiveness by a margin that is comparable to successive recent publications on this dataset. Our technique is completely unsupervised, which means that the gains come essentially for free. We confirm that the improvements can be directly attributed to our heuristic, as other approaches to extracting comparable amounts of training data are not effective. Beyond the empirical validation of this heuristic, we also share our improved TrecQA dataset with the community to support further work in answer selection.

1 INTRODUCTION

There are three key components to solving problems with machine learning: the training data, the model, and the optimization technique. To improve effectiveness, data is often the easiest path since in some applications it is easy to collect a large amount of data, such as user behavior logs in the web context. In contrast, improving models and optimization techniques often require inspiration.

In this paper, we focus on the data dimension of improving answer selection for question answering. We propose a heuristic that we call "one answer per document", which yields a simple technique for extracting high-quality negative examples. Starting with question–answer pairs from the popular TrecQA dataset, one of the most widely-used collections for evaluating answer selection in question answering, we identify the original documents from which the answers are drawn. The best-matching sentences from these source documents that contain query terms (other than the answer sentences) are selected as negative examples. Training on the original data plus these negative examples yields improved effectiveness. Our intuition is that the answer to a question is only likely to occur once in a document, and thus other sentences in the document with query terms can serve as high-quality negative examples. We argue that these examples are particularly valuable because they lie near the decision boundary (by virtue of containing query terms). This intuition is confirmed by contrastive experiments that show alternative techniques for acquiring comparable amounts of training data are not effective.

The contributions of this work are two-fold: First, we propose and empirically validate the effectiveness of the "one answer per document" heuristic. Our approach is completely unsupervised, which means that gains in effectiveness come with minimal effort. Examining the history of improvements on this task, the gain we achieve is around the same level of effectiveness as reported in successive recent publications on this dataset, nearly all of which come from improved modeling using neural networks. Second, our technique yields an improved and augmented version of the widely-used TrecQA dataset that we share with the community to foster further work on answer selection.

SIGIR'17, August 7–11, 2017, Shinjuku, Tokyo, Japan

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-5022-8/17/08…$15.00

DOI: http://dx.doi.org/10.1145/3077136.3080645

2 BACKGROUND AND RELATED WORK

Answer selection is an important component of an overall question answering system: given a question q and a candidate set of sentences $\{s_1, s_2, \ldots s_n\}$, the task is to identify sentences that contain the answer. In a standard pipeline architecture [12], answer selection is applied to the output of a module that performs passage retrieval, typically using lightweight term-based matching. Selected sentences can then be directly presented to users or serve as input to subsequent stages that identify exact answers [13].

In recent years, researchers have had substantial success in tackling the answer selection problem with neural networks, e.g., [6, 7, 10, 11, 17]. The continuous representations that deep-learning approaches provide are effective in combating data sparsity, a perpetual challenge in natural language processing tasks. Solutions based on neural networks represent an advance over previous approaches driven by feature engineering. Although our work is primarily about techniques for acquiring training data, we assume a deep-learning framework for evaluation purposes.

It is a well-known fact that the amount of training data drives effectiveness in a broad range of tasks. Colloquially referred to as the "unreasonable effectiveness of data" [5], researchers have been empirically examining the impact of training data for machine learning since at least the early 2000s. The seminal work of Banko and Brill [2] in examining the effects of training data size on natural language disambiguation tasks contained a slightly subversive message, that the effort of researchers might be better spent gathering

Dataset	Document Collections
TREC8	TREC disks 4&5 minus Congressional Record
TREC9 TREC10	AP newswire (Disks 1-3) Wall Street Journal (Disks 1-2) San Jose Mercury News (Disk 3) Financial Times (Disk 4) Los Angeles Times (Disk 5) Foreign Broadcast Information Service (FBIS) (Disk 5)
TREC11 TREC12 TREC13	AQUAINT disks

Table 1: Source document collections for TrecQA.

training data as opposed to building more sophisticated models. Specifically in the realm of question answering, researchers have long known that, all things being equal, larger collections yield higher effectiveness due to data redundancy [3, 4].

In this context, our work focuses on gathering training data for answer selection in order to improve machine-learned models. Specifically, our "one answer per document" heuristic is a nod to the "one sense per discourse" heuristic Yarowsky [16] applied to word-sense disambiguation, which dates back to the 1990s. This work is an early example of a clever technique for acquiring (noisy) labeled data for free, much like our work. The intuition behind the heuristic is that polysemous words occurring close together are unlikely to have different senses. For example, if the word "bank" occurs in nearby sentences, it is unlikely that one refers to a financial institution and the other to the side of a river. This is an artifact of how authors naturally communicate when writing. From this heuristic Yarowsky described an approach to bootstrap a word-sense disambiguation algorithm. In the same way, our "one answer per document" heuristic reflects how authors write. There can, of course, be violations of this heuristic, but the point is that sufficient signal can be extracted with this heuristic to aid in training machine-learned models. Our technique is closely related to what researchers today would call distant supervision, but our focus is specifically on data acquisition.

3 METHODS

In order to operationalize our "one answer per document" heuristic, we build on the TrecQA dataset that is broadly used as a benchmark for answer selection. Note that although this paper focuses on a specific dataset—since one of our contributions is a resource we share with the community—the assumptions we make about the general technique are fairly minimal: simply that answer sentences are drawn from documents within some collection.

The TrecQA dataset was first introduced by Wang et al. [14] and further elaborated by Yao et al. [15]. The dataset contains a set of factoid questions, each of which is associated with a number of candidate sentences that either contain or do not contain the answer (i.e., positive and negative examples). The questions are from the Question Answering Tracks from TREC 8–13, and the candidate answers are derived from the output of track participants, ultimately drawn from the collections listed in Table 1.

The TrecQA dataset comes pre-split into train, development, and test sets, with statistics shown in Table 2. Questions from TREC 8–12 are used for training (1229 questions), while questions from TREC 13 are used for development (84 questions) and testing (100 questions). To generate the candidate answers for the development and test splits, sentences were selected from each question's evaluation pool that contained one or more non-stopwords from the question [14]. For generating the training candidates, in addition to the sentences that contain non-stopwords from the question, sentences that match the correct answer patterns (from an automatic evaluation script) were also added. Data from all of TREC 13 (development and test splits) and the first 100 questions from TREC 8–12 (training split) were manually assessed. The motivation behind the manual annotation effort is that answer patterns in the automatic evaluation script may yield false positives—i.e., sentences that match the pattern may not actually contain correct answers.

Set	# Question	# Pos Answers	# Neg Answers
Train	1,229	6,403	47,014
Dev	84	222	926
Test	100	284	1,233
All	1,411	6,909	49,173

Table 2: Statistics for various splits of TrecQA.

Although the TrecQA dataset was ultimately constructed from TREC evaluations, the provenance information connecting answer candidates to their source documents does not exist. Therefore, to operationalize our "one answer per document" heuristic, we needed to "backproject" each answer candidate to recover its source document. Note that due to tokenization, case folding, and other sentence processing differences, finding the answer sentence is more complex than just an exact string match.

Answer backprojection was accomplished by first indexing all the collections in Table 1 with Anserini,[1] our information retrieval toolkit built on Lucene. We then issued each question as a query and retrieved the top 1000 hits using BM25. For each answer a, we used the shingle matching method [8, 9] to select the most likely candidate document d that contains the answer a. For an answer a, let s be the minimum span of words in a candidate document d that contains the most words from a in any order. A span s matches a well if s contains many words from a within a small window. We used the algorithm presented by Krenzel [8] to find the shortest span s of shingle words within a document in linear time:

$$Score_{s \in d} = \max \frac{|s \cap a|^2}{|s| \cdot |a|} \tag{1}$$

After we find the best matching document d for an answer a, we split the sentences in d using the NLTK Punkt sentence tokenizer.[2] Equation (1) is used again to score all sentences in d; we take as the matching answer the highest scoring sentence above a threshold of 0.1. If no sentence scores above this threshold, we drop the answer from consideration. Based on spot-checking, this setting is able to find the source sentence with nearly perfect precision. Once we have found the source sentence, all other non-zero scoring sentences in the document provide negative examples for the answer selection task, based on our "one answer per document" heuristic.

[1]http://anserini.io/
[2]http://www.nltk.org/api/nltk.tokenize.html

Question		Who is the author of the book , " The Iron Lady : A Biography of Margaret Thatcher " ?
Answer		the iron lady ; a biography of margaret thatcher by hugo young -lrb- farrar , straus & giroux -rrb-
Ranking	Score	Sentence
Matching	0.681	THE IRON LADY: A BIOGRAPHY OF MARGARET THATCHER BY HUGO YOUNG (FARRAR, STRAUS &amp; GIROUX: $25; 570 PP.
1	0.244	In "The Iron Lady," Young traces the winding staircase of fortune that transformed the younger daughter of a provincial English grocer into the greatest woman political leader since Catherine the Great.
2	0.208	It is without question the best of a bevy of new Thatcher biographies that set out the often surprising, always dramatic story of the British political revolution of the 1980s.
3	0.203	In this same revisionist mold, Hugo Young, the distinguished British journalist, has performed a brilliant dissection of the notion of Thatcher as a conservative icon.
4	0.195	The implied paradox has been nicely captured by a recent British assessment of the last six years titled "The Free Economy and the Strong State: The Politics of Thatcherism" by Andrew Gamble.
5	0.133	It sees Thatcher as the new Metternich (the 19th-Century master of the diplomatic finesse), as a power-driven politician and as a militant Puritan.
6	0.130	Young observes that "There was a genuine clash of cultures, between an almost Cromwellian impatience with the status quo (on the part of the Thatcherites) and the mandarin world of Whitehall, in which skepticism and rumination were more highly rated habits of mind than zeal or blind conviction."
7	0.125	The only company nominated by Thatcher's team for denationalization was the National Freight Corp., and this from the people who much later made "privatization" one of the household words of the age.

Table 3: Example backprojection of an answer to recover the source document (LA111289-0002) and the source sentence. Non-matching sentences serve as negative examples.

A complete example of this backprojection process is shown in Table 3. At the top we show the question and the answer we are trying to find. First, we identified document LA111289-0002 as the source document. In this document, in addition to the top-scoring sentence (a correct match), we also show the non-matching sentences in decreasing score order. This example illustrates why finding the source answer requires more than an exact string match. The non-matching sentences show the intuition behind our "one answer per document" heuristic—indeed, none of the sentences answer the question. Note that although sentence 3 contains the author "Hugo Young", it doesn't provide any contextual justification, i.e., there is no way for the reader to infer the answer in isolation. Accordingly, it should be considered a negative example.

4 EVALUATION AND RESULTS

We applied the procedure described above to backproject answer sentences from the TrecQA dataset to reconstruct their sources. Operationalizing the "one answer per document" heuristic, non-matching sentences from the source document containing the answer serve as negative examples we can use to augment the training data. We considered cases where $m \in \{1, 3, 5, 7\}$ of these top non-matching sentences are added to the training set as negative examples (see Table 3). We tokenized these sentences using the standard Penn Treebank format[3] to match the original dataset.

How effective is the "one answer per document" heuristic? To find out, we trained an answer selection model using our augmented training data and compared the results with training on the original data. For this task, we used the convolutional neural network model of Severyn and Moschitti [11] (SM for short). Their model achieves competitive accuracy and the authors provide an open-source implementation.[4] Following previous work, we evaluated the task in terms of Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). In order to train the model more efficiently, for negative sentences selected by our technique, we truncated their lengths to 60 tokens. All parameters and the training procedure remained the same as in the original model. We emphasize that in all our experiments, the only difference is an augmented training set: the development set and test set remain exactly the same, thus supporting a fair comparison of results. Any difference in effectiveness can be directly attributed to the training data.

Results of this experiment are shown in Table 4. The row labeled "Baseline SM Model" is the result of our replication using the implementation provided by Severyn and Moschitti and in fact we obtain slightly higher effectiveness than what they reported. The next four rows in the table show the effects of adding different numbers of negative examples per each backprojected answer. For example, with $m = 5$, we would add up to $6{,}403 \times 5 = 32{,}015$ negative examples. We see that the $m = 1$ condition reduces effectiveness slightly, likely due to the noise introduced. Adding more sentences helps, peaking at $m = 5$, and then effectiveness drops again.

The intuition behind our "one answer per document" heuristic is that our data acquisition algorithm yields high-quality negative examples that are valuable because they lie near the decision boundary. Experimental results support this claim, but to further validate our heuristic, there are two alternative explanations to rule out: First, that these sentences might be even more useful as positive examples, and second, that the gains aren't derived from simply having more training data.

To explore the first alternative explanation, we repeated the same experiment as above, augmenting the training set with $m \in \{1, 3, 5, 7\}$ of the top non-matching sentences, but as *positive* examples. Results are also shown in Table 4. We clearly see that such a treatment hurts effectiveness for all examined values of m. This finding is consistent with the assumption that the answer will only appear once in each document, thus supporting our heuristic.

To explore the second alternative explanation, we experimented with two different approaches to augmenting the training set: in the first case, we selected five random sentences from the answer document to serve as negative examples (and thus, they may or may not contain terms from the question), and in the second case, we randomly selected five sentences from all documents to serve as

[3] http://www.nltk.org/_modules/nltk/tokenize/treebank.html
[4] https://github.com/aseveryn/deep-qa

Strategy	MAP	MRR
Baseline SM Model	0.7538	0.8078
Add top 1 sent as neg	0.7468	0.7953
Add top 3 sent as neg	0.7588	0.8012
Add top 5 sent as neg	**0.7612**	**0.8088**
Add top 7 sent as neg	0.7493	0.7993
Add top 1 sent as pos	0.7409	0.7974
Add top 3 sent as pos	0.7193	0.7670
Add top 5 sent as pos	0.7117	0.7456
Add top 7 sent as pos	0.7016	0.7639
Add random 5 neg sent from correct documents	0.7548 [0.7499, 0.7597]	0.8075 [0.8038, 0.8112]
Add random 5 neg sent from all documents	0.7526 [0.7496, 0.7556]	0.7969 [0.7883, 0.8054]
Multi-Perspective CNN	0.762	0.830
Add top 3 sent as neg	**0.7864**	**0.8325**
Add top 5 sent as neg	0.7788	0.8316

Table 4: Results of comparing different strategies.

negative examples. We conducted five trials of each experimental condition so that we can compute the mean and 95% confidence intervals for both MAP and MRR. These results are also shown in Table 4. We see that both sampling approaches have minimal effect on effectiveness (to be expected). Note that in these cases the neural network is trained with the same amount of data as in the negative sampling case. Combined with the above experiments, these results confirm that effectiveness gains do not come from simply having more data, but having high-quality *negative* examples, thus supporting our "one answer per document" heuristic.

Let us tackle the next possible criticism: that we are improving on a low baseline. As a point of reference, we can consult an ACL wiki page that nicely summarizes the state of the art in this answer selection task [1]. We clearly see that while the SM model isn't the top-performing model on this task, its effectiveness remains competitive. To show the robustness of the effectiveness gains that we observe, we also experimented with the multi-perspective convolutional neural network (MPCNN) architecture of He et al. [6], which also has open-source code available.[5] Since this model is more complex than the SM model and hence takes longer to train, we only repeated the condition of adding $m \in \{3, 5\}$ negative examples. Results show gains in both conditions, with $m = 3$ appearing to be the better setting.

Finally, let us try to contextualize the magnitude of gains that derive from our technique. The ACL wiki page [1] provides the history of effectiveness improvements over time. Of course, in the beginning right after this dataset was published, researchers made great strides in improving effectiveness. However, the magnitude of advances has dramatically shrunk: in recent years, publications are reporting small gains in the second decimal point. All of these improvements are from increasingly-sophisticated neural network models. The magnitude of our observed improvements is comparable to differences in successive recent publications on this particular dataset: for example, the improvement from He et al. [6] (published in 2015) to Rao et al. [10] (published in 2016) is less than 0.02 in terms of absolute MAP. The magnitude of our gains is at least as large, and in fact, our best condition appears to be the highest reported result on TrecQA (as of this writing). Since our technique is completely unsupervised, these gains basically come for free.

As a resource for the community, we release all data from this paper, including the source document mappings and the negative examples to augment the original TrecQA dataset.[6]

5 CONCLUSIONS

Data, model, and optimization represent three different approaches to increasing the effectiveness of machine learning solutions. This paper adopts the data approach to tackling answer selection: We begin with an intuition, the "one answer per document" heuristic, that we then operationalize into a data acquisition algorithm. Augmented training data improves the effectiveness of existing models, and contrastive experiments rule out alternative explanations for our findings, thus validating our approach. As applied to a specific dataset, the widely-used TrecQA benchmark, our work yields an improved data resource that we share with the community. However, we believe that this heuristic is equally applicable to other tasks and datasets, a future direction that we are currently pursuing.

REFERENCES

[1] ACL. 2017. Question Answering (State of the art). http://www.aclweb.org/aclwiki/index.php?title=Question_Answering_(State_of_the_art). (2017). Accessed: 2017-05-01.
[2] Michele Banko and Eric Brill. 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In *ACL*. 26–33.
[3] Charles L. A. Clarke, Gordon Cormack, and Thomas Lynam. 2001. Exploiting Redundancy in Question Answering. In *SIGIR*. 375–383.
[4] Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. 2002. Web Question Answering: Is More Always Better? In *SIGIR*. 291–298.
[5] Alon Halevy, Peter Norvig, and Fernando Pereira. 2009. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems* 24, 2 (2009), 8–12.
[6] Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks. In *EMNLP*. 1576–1586.
[7] Hua He and Jimmy Lin. 2016. Pairwise Word Interaction Modeling with Neural Networks for Semantic Similarity Measurement. In *NAACL-HLT*. 937–948.
[8] Steve Krenzel. 2010. Finding blurbs. http://www.stevekrenzel.com/articles/blurbs.
[9] Virgil Pavlu, Shahzad Rajput, Peter B. Golbus, and Javed A. Aslam. 2012. IR System Evaluation using Nugget-based Test Collections. In *WSDM*. 393–402.
[10] Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-Contrastive Estimation for Answer Selection with Deep Neural Networks. In *CIKM*. 1913–1916.
[11] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *SIGIR*. 373–382.
[12] Stefanie Tellex, Boris Katz, Jimmy Lin, Gregory Marton, and Aaron Fernandes. 2003. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In *SIGIR*. 41–47.
[13] Ellen M. Voorhees. 2002. Overview of the TREC 2002 Question Answering Track. In *TREC*.
[14] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *EMNLP-CoNLL*. 22–32.
[15] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer Extraction as Sequence Tagging with Tree Edit Distance. In *HLT-NAACL*. 858–867.
[16] David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *ACL*. 189–196.
[17] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep Learning for Answer Sentence Selection. In *NIPS Deep Learning Workshop*.

Acknowledgments. This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and by a Google Founders Grant, with additional contributions from the U.S. National Science Foundation under CNS-1405688.

[5]https://github.com/castorini/MP-CNN-Torch

[6]https://github.com/castorini/TrecQA-NegEx