Improved Similar Question Retrieval through LSTM Semantic Embeddings and Siamese Network

Heng Lin
University of Amsterdam
henry.lin@student.uva.nl

Joop Pascha
University of Amsterdam
joop.pascha@student.uva.nl

Luca Simonetto
University of Amsterdam
luca.simonetto@student.uva.nl

Muriël Hol
University of Amsterdam
muriel.hol@student.uva.nl

Abstract

introduction to the problem. contributions in terms of improvements on previous implementation. comparison of results to start-of-the-art.

Keywords— *todo keywords*

1 Introduction

The discipline of Question Answering (QA) is concerned with systems that can automatically answer questions posed by humans in natural language. Today, a large set of questions that regular users have posed on websites such as stackoverflow and yahoo answers have already been answered which could potentially be used to answer similar questions directly. This poses the advantage of avoiding *question-starvation* in which some questions are never answered or take a significant time. However, finding similar questions requires a thorough understanding of the semantic meaning thereof. The *lexicon-syntactic gap* poses a significant problem as questions can be significantly different lexically and syntactically while having the same semantic meaning. Here, an attempt is made to uplift this limitation with the use of Siamese networks in a similar fashion to Das et al. [3] with a change of input representation and network structure.

Das et al. [3] propose the use of Siamese networks to resolve both the problem of finding an appropriate embedding that projects similar questions close to each other in semantic space while at the same time solving the issue of having no question-to-question dataset that is sufficiently large. The latter is important as semantic embeddings generally require an abundance of data. They propose an unsupervised training method with state-of-the-art results in which question-answer pairs (Q,A) are used to train a model of which only the semantic embeddings are used to compare (Q,Q) similarity during testing. This is achieved is by exploiting the weight-sharing property of the Siamese network with a contrastive loss function that tries to bring the two inputs matching (Q,A) pairs closely together in semantic space while using (Q,A)'s that do not belong to each as negative examples which are pushed further apart. Hence the name contrastive loss function. We believe that an other benefit of this approach compared to using question-to-question pairs only, is that answers tend to be longer than questions and therefore indirectly encapsulate more of the semantic meaning otherwise hidden for algorithms within the question.

In this paper we use a Siamese Long-Short-Term-Memmory (LSTM) with contrastive loss function to find questions that are semantically similar to each other in order to further remedy the lexicon-symantic gap and question-starvation. In addition, an other Siamese LSTM is used for topic modeling of which features are extracted and used to further enhance the LSTM with contrastive loss function. The Yahoo[1] dataset is used that contains Q-A pairs with topic meta-data. We hypothesize that this yield improvements over the *bag-of-words* tri-gram representation of Das et al. [3] for two reasons. First, by using a modeling approach that takes words and context into account it is expected that a more semantically rich input representation is achieved. Second, by using a hybrid approach that combines two of the most recent QA approach; deep learning (LSTM) with topic modelling, results are expected to be boosted. For comparison, Das et al. [3] work is replicated and compared to the results of our approach.

We identify the contributions of our paper as followings:

- First application of Siamese LSTM's to the QA domain for similar questions retrieval
- Use instead of sentence-sentence input data, paragraph-paragraph or sentence-paragraph representation
- Fusing Topic Models with Deep Learning Based Approaches

The remainder of this paper is broken up in the following sections. In Related Work (2) the approach of Das et al. [3] is explained in more detail and the relevant topics which we use in our approach such as Siamese networks and LSTM are explained more thoroughly. Thereafter, in Methods (4) our adaptations and enhancements are explained, including a detailed view of the data processing steps and network architecture. The performance of the models are then shown in Results (7). For this we use qualitative methods (MAP, MRR, P@K) and quantitative analysis on an annotated test set [12]. An overview of our most important findings and remarks are then discussed in the conclusion (8) after which

[1] Yahoo! Answers Comprehensive Questions and Answers version 2.0, available here

we give our recommendation for future work (9).

2 Related Work

The current state of the art in QA appears to go into the direction of deep learning approaches and topic models. For an overview of the history of question answering we defer to Das et al. [3]. Instead, here we focus first on given a formal definition of the Siamese twin network which is consequently used in Methods 4. Second, an explanation of the use of LSTMs is being given and how this could potentially be used as a feature extractor for the Siamese network. Third, a short explanation is given on the deep learning methods within Natural Language Processing and how this knowledge can be used to enhance the approach of Das et al. [3].

Siamese networks are neural networks that consist of two identical sub-networks joined at their output [2]. Identical means here that both share the architecture as well as the model-parameters. In the most basic form, the network can be seen as having two inputs x_1 and x_2 which are fed to the same network twice after which a loss function is applied that compares the similarity between them, see figure **todo**. Bromley et al. [2] first proposed this idea and showed that this method could be used for signature verification (computer vision) by setting a threshold on the similarity to authentic and unknown signatures. An argument is made for using low-level features as input such that the network is able to learn high-order features themselves. The most significant differences to Das et al. [3] is the use of a contrastive loss function which allows for supervised learning and it's applicability to Natural Lanuage Processing.

The contrastive loss function not only enforces similar items to be close in semantic space, but also that dissimilar items are further apart. Das et al. [3] show that this can also be applied to the QA domain in which question-answer pairs can be used as input of the model, x_1 and x_2, after which a contrastive loss function can applied that uses a cosine similarity metric to model output representation; q_i and a_i. For this, the loss function is conditioned on the label that depicts whether the question-answer came from the same (1) or different questions (-1) which either causes the semantic representation of x_1 and x_2 to contract or diverge respectively. The importance of the latter can be determined by parameter m that is defined between $[0, 1]$ such that lower values of m places more emphasis on dissimilar items being far apart in semantic space.

$$contractive\_loss = \begin{cases} 1 - cos(q_i, a_i) & \textit{if y = 1} \\ max(0, cos(q_i, a_i) - m) & \textit{if y = -1} \end{cases}$$

Paragraph describing the method of Das et al. [3].

While Das et al. [3] adhered to a low-level 3-grams character Boolean input vector representation similar to Bromley et al. [2], there are also Siamese approaches that use more complex word-presentation in combination with using Convolutional Neural Nets [5, 7, 11] or Long-Short-Term-Memory (LSTM) [8] with success. The combination of *bag-of-words* combined with convolutional neural networks as apposed in Das et al. [3] seem to leave room for improvements. Therefore, an approach more similar to Mueller and Thyagarajan [8] is taken in which an LSTM is used for modeling sentence-sentence similarity. However, there are two noteworthy differences. First, we apply the approach to a different domain (QA). Second, instead of using sentences, both the questions and answers can be modeled as sequences of sentences. The does not seem to consider this temporal aspect due to their bag of words assumption. Although deep learning approaches are slowly adapted to the field of Natural Language Processing, we focus here instead on the current state-of-the art in modeling context dependent sequences; LSTMS.

The weight-sharing properties of Siamese networks are essential Weight sharing across time-sequences is crucial for two reasons [4]. First, it enables the possibility of using an arbitrary sequence length. Second, it is now possible to use the same transition function and parameters at every timestep which greatly reduces the dimensionality of the problem.

Recurrent Neural Networks (RNN) pose the benefits of re-using the same parameters across time and being able to handle any sequence length in theory. While vanilla RNN's had some significant problems due to their exploding and vanishing gradients for longer time sequences, LSTM's partially resolves this problem.

3 Task Definition

4 Methods

Data

For collecting the training- and validation-set 2 and 0.4 million unique examples were sampled from the Yahoo Labs Webscope: *L6 - Yahoo! Answers Comprehensive Questions and Answers version 1.0 (multi part)* [2]. The dataset contains in total 4483032 questions (some of which do not have answers). One example contains the meta-data that can be seen in Table 1. For simplicity only the *bestanswer* was selected such that each *uri* could be used within our dataset to map back to the original and unique Q-A pair. It was expected that the *bestanswer* was the closest in semantic meaning of the question and that possibly a bias towards question with multiple answers were obtained if all answers (or subset) was considered. However, for negative examples which come from a Q-A pair of which the answer does not belong to the question (thus have different *uri's*), the *uri* was extended for our dataset to denote $uri_Q\_uri_A$ which for positive examples are the same ($uri_Q == uri_A$).

As the Yahoo dataset is around 12 GB unparsed with 4483032 unique *uri*'s a sampling method was required. Only *uri*'s with the qlang tag set to english and that included at least one of the category tags *cat*, *main* or *subcat* were included, resulting in **todo** unique questions.

For collecting of the test-set the the labeled dataset from Zhang et al. [12] is used which consists of 1018 questions. This dataset is already used **todo, after finding out what the test-set actually consists of describe what it consists of and explain how learning to rank is used here.**.

-

[2] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l

***T1:** Most important tags for our purpose within the Yahoo dataset*

xml tag	description	optional	who?
uri	unique resource identifier	no	yahoo
subject	abbreviated question	yes	questioner
content	added context to the question, subject is repeated if not specified	no	questioner
bestanswer	questioner selects best answer within a predetermined amount of time, afterwards the upvotes of the community determine the best answer.	no	questioner/community
answer_item	a question can have multiple answers, denoted by answer_item	no	community
cat	each question is automatically assigned a cat after being posted	yes	yahoo
maincat	each question is automatically assigned a maincat after being posted	yes	yahoo
subcat	each question is automatically assigned a subcat after being posted	yes	yahoo
qlang	language of question	no	yahoo

Input Representation

For the 3-grams character representation used by Das et al. [3] a binary vector was created with the *bag-word-words* assumption. For this, all unique 3-grams in the training set were listed and hashed to a unique index within the binary vector. Consequently each question and answer were multi-hot-encoded.

For the input of our LSTM pre-trained word embeddings were used. Jastrzebski [6] ran multiple state-of-the-art of which an implementation is publicly available on the internet for a variety of different benchmarks. The *LexVec* implementation word *word2vec* was used here based on this result of which the code is based on Salle et al. [9, 10] and available at github.

Data Pre-processing

For the 3-grams character representation each question and answer was pre-processed by lower-casing, stemming, stop-word and special character removal. For the stopwords and stemming the library Natural Language ToolKit (NLTK) was used [1].

For the input of our LSTM sentences were first parsed which were then tokenised as a necessary pre-processing step for the *LexVec* library. **todo**.

Network Architectures

For the 3-grams character representation the same architecture was used as in Das et al. [3]. However, due to ambiguity and lack of response of the authors a few design design decisions were made. First, the binary vector is assumed to first be connected to a fully connected layer of the same size which is thereafter connected to the fully connected layer and fully connected layer. Confusion arose in the authors figure 2 where W_1 is displayed between the vector representation of question/answer, but there is no direct mention thereof in section 3. From a practical point of view it makes no sense to connect an uncorrelated sparse *bag-of-words* input directly to a convolutional layer, which stresses the necessity of this layer.

Second, the assumption was made that in their Table 1 the *Depth of CNN* refers to the number of consecutive Convolutional layers, which could correct the misplaced weights in the same figure 2 for Max Pooling (W_3) and ReLu layer (W_4). Instead here we take these as referring to Convolutional Layers. As no *filter*, *strides*, or *padding* was specified, some experiments were run for optimal performance on the validation set. The model performance under different settings can be seen in table **todo** of which the final architecture can be seen in figure **todo**. For more information on model training, see the Model Training section 5.

For the LSTM representation, the necessity for hierarchical LSTM was proven important **verify if this is going to be true** due to questions generally tending to be only one or a few sentences, while answer tend to be significantly longer. Therefore a an LSTM for required for sentence representation and paragraph representation. Data was collected to show this phenomena of different sentence/word for question and answer which is shown in figure **todo**.

Loss Function

For both models the same contrastive loss was used as introduced by Das et al. [3]. For completeness these are shown below, for a more detailed explanation of parameters of the $contractive\_function$ we defer back to the Related Work section.

$$loss(q_i, a_j) = \begin{cases} 1 - cos(q_i, a_j) & i = j \\ max(0, cos(q_i, a_j) - m) & i \neq j \end{cases} \qquad (1)$$

Meaning that cosine similarity is attempted to be maximized for question and answering pairs that belong together (positive examples: $i = j$), while dissimilarity is attempted to be maximized for answers that do not belong to the question (negative examples: $i \neq j$).

$$\mathcal{L}(\Lambda) = \sum_{(q_i, a_j) \in C \cup C'} loss(q_i, a_j) \qquad (2)$$

Here, the loss is minimized per batch that consists of an equal proportion of positive, C, and negative examples, C'. The Stochastic Gradient Descent (SGD) optimizer is used.

5 Model Training

6 Evaluation

Quantitative Analysis

Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Precision at K (P@K)

Qualitative Analysis

7 Results

8 Conclusion

9 Future Work

References

[1] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[2] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

[3] Arpita Das, Harish Yenala, Manoj Chinnakotla, and Manish Shrivastava. Together we stand: Siamese networks for similar question retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 378–387, 2016.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[5] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050, 2014.

[6] Stanislaw Jastrzebski. word-embeddings-benchmarks wiki. `https://github.com/kudkudak/word-embeddings-benchmarks/wiki`. (Accessed on 11/18/2017).

[7] Zhengdong Lu and Hang Li. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*, pages 1367–1375, 2013.

[8] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792, 2016.

[9] Alexandre Salle, Marco Idiart, and Aline Villavicencio. Enhancing the lexvec distributed word representation model using positional contexts and external memory. *arXiv preprint arXiv:1606.01283*, 2016.

[10] Alexandre Salle, Marco Idiart, and Aline Villavicencio. Matrix factorization using window sampling and negative sampling for improved word representations. *arXiv preprint arXiv:1606.00819*, 2016.

[11] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*, 2015.

[12] Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 371–380. ACM, 2014.