

Ways of Asking and Replying in Duplicate Question Detection

João Rodrigues, Chakaveh Saedi, Vladislav Maraev, João Silva, António Branco

University of Lisbon

{joao.rodrigues,chakaveh.saedi,vlad.maraev,
jsilva,antonio.branco}@di.fc.ul.pt

Abstract

This paper presents the results of systematic experimentation on the impact in duplicate question detection of different types of questions across both a number of established approaches and a novel, superior one used to address this language processing task. This study permits to gain a novel insight on the different levels of robustness of the diverse detection methods with respect to different conditions of their application, including the ones that approximate real usage scenarios.

1 Introduction

Automatic detection of **semantically equivalent questions** is a language processing task of the utmost importance given the upsurge of interest in conversational interfaces. It is a key procedure in finding answers to questions. **For instance**, in a context of **customer support via a chat channel**, with the help of duplicate question detection, previous interactions between customers and human operators can be explored to provide an increasingly automatic question answering service. If a new input question is equivalent to a question already stored, it can be replied automatically with the answer stored with its recorded duplicate.

Though it has been less researched than **similar tasks**, **duplicate question detection (DQD)** is attracting an increasing interest. It can be seen as belonging to a **family of semantic text similarity tasks**, which have been addressed in **SemEval challenges** since 2012, and which in the last SemEval2016, for instance, included also tasks like plagiarism detection or degree of similarity between machine translation output and its post-edited version, among others. **Semantic textual similarity** assesses the degree to which two tex-

Semantic equivalence is a declaration that two data elements from different vocabularies contain data that has similar meaning.

tual segments are semantically equivalent to each other, which is typically scored on an ordinal scale ranging from semantic equivalence to complete semantic dissimilarity.

Paraphrase detection can be seen as a special case of semantic textual similarity, where the scale is reduced to its two extremes and the outcome for an input pair is yes/no. DQD, in turn, could be seen as a special case of paraphrase detection that is restricted to interrogative expressions.

While SemEval2016 had no task on yes/no DQD, it had a “Question-Question” graded similarity subtask of Task 1. The top performing system in this subtask (0.74 Pearson correlation) scored below the best result when all subtasks of Task 1 are considered (0.77), and also below the best scores of many of the other subtasks (e.g. 0.84 in plagiarism detection (Agirre et al., 2016)).

While scores obtained for different tasks by systems trained and evaluated over different datasets cannot be compared, those results nonetheless lead one to ponder **whether focusing on pairs of interrogatives may be a task that is harder than paraphrase detection that focuses on pairs of non-interrogatives** (e.g. plagiarism pairs), or at least whether it needs different and specific approaches for similar levels of performance to be attained.

When checking for other research results specifically addressing DQD, pretty competitive results can be found, however, as in **Bogdanova et al. (2015)**. These authors used a dataset that included a dump from the Meta forum in Stack-Exchange (a source that would be explored also in SemEval2016) and a dump from the AskUbuntu forum, and reported over 92% accuracy.

The pairs in these datasets are made of the textual segments that are submitted by the users of the forums to elicit some feedback from other users that may be of help, and that will pile up in threads of reactions. They have two parts, known as “ti-

DATASET

tle” and “body”. The title tends to be a short segment identifying the issue being addressed, and the body is where that issue is expanded, and can be several paragraphs long.

To avoid a maze of exactly duplicate questions, and thus of duplicate threads, which would hamper the usability of the forums, for the same issue, all duplicates except one are removed, leaving only near duplicates—that are marked as such and cross-linked to each other, and may be of help in addressing the same topic from a different angle.

The pairs of duplicate segments included in the experimental datasets mentioned above are the titles and bodies of nearly duplicate threads. The pairs of non-duplicate segments are made of titles and bodies that are not near duplicate.

While these “real life” data are important for the development of DQD solutions that support the management of these community forums, their textual segments are quite far from expressions in clean and clear interrogative form. The short supply of this sort of datasets has been perhaps part of the reason why the DQD has not been more researched. This may help to explain also the lack of further studies so far on how the nature of the questions and the data may impact the performance of the systems on this task.

The experiments reported in this paper aim to address this issue and help to advance our understanding of the nature of DQD and to improve its application. We will resort to previous datasets used in the literature, just mentioned above, but we will seek to explore also a new dataset from Quora, released recently, in January 2017.

The pairs of segments in this Quora dataset concern any subject and are thus not restricted to any domain. The segments are typically one sentence long, clean and clear interrogative expressions. Their grammatical well-formedness is ensured by the volunteer experts that answer them and that, before writing their replies, can use the editing facility to adjust the wording of the question entered by the user if needed.

This is in clear contrast with the other datasets extracted from community forums. The forums are organized by specific domains. The segments may be several sentences long and are typically offered in a sloppy wording, with non-standard expressions and suboptimal grammaticality.

By resorting only to data of the latter type, Bogdanova et al. (2015) confirmed that systems

trained (and evaluated) on a smaller dataset that is domain specific can perform substantially better than when they are trained (and evaluated) on a larger dataset from a generic domain.

In this paper, we seek to further advance the understanding of DQD and possible constraints on their development and application. We assess the level of impact of the length of the segments in the pairs, and study whether there is a difference when systems handle well-edited, generic domain segments, versus domain specific and sloppy ones.

As the datasets with labeled pairs of segments are scarce, to develop a system to a new specific domain lacking a training dataset, the natural way to go is to train it on a generic domain dataset. We also study the eventual loss of performance in this real usage scenario.

These empirical contrasts may have a different impact in different types of approaches to DQD. The present study will be undertaken across a range of different techniques, encompassing a rule-based baseline, a classifier-based system and solutions based on neural networks.

To secure comparability of the individual results, the experimental datasets used are organized along common settings. They have the same volume (30K pairs), the same training vs. testing split rate (80%/20%), and the same class balance (50%/50% of duplicates and non-duplicates).

This paper is organized as follows. In Section 2, the datasets used are described. Sections 3, 4 and 5 present the experimental results of a range of different detection techniques, respectively, rule-based, supervised classifiers and neural networks. In section 6, the results obtained are discussed, and further experiments are reported in Section 7, approximating a real usage scenarios of application. Sections 8 and 9 present the related work and the conclusions.

2 Datasets

We used two datasets, from two sources:¹ (i) from the AskUbuntu online community forum where a query entered by a user (in the form of a title followed by a body) is answered with contributions from any other user (which are piled up in a thread); and (ii) from Quora, an online moderated question answering site where each query introduced by a user, typically in a grammatical inter-

¹Datasets are available from <https://github.com/nlx-group/dqd>

1	Q How is the new Harry Potter book 'Harry Potter and the Cursed Child'?
	Q How bad is the new book by J.K Rowling?
0	Q Should the toothbrush be wet or dry before applying the toothpaste?
	Q What is the cheapest toothpaste?
1	Q Can I install Ubuntu and Windows side by side?
	Q How do I dual boot Windows along side Ubuntu?

Figure 1: Three example question pairs and their labels from the Quora dataset

1	Q Why is more than 3GB of RAM not recognised when using amd64?
	Q Ubuntu 10.04 LTS 64bit only showing 2.9GB of memory
1	Q How can I fix a 404 Error when updating packages?
	Q What does this mean & what impact does it have: Failed to download repository information
0	Q hipshop, nginx, spdy
	Q print xlsx file from command line using ghostscript and libreoffice

Figure 2: Three example segment pairs (titles only) and their labels from the AskUbuntu dataset

rogative sentence, receives an answer often from a volunteer expert. For either dataset, the language of the textual segments is English.

We resorted to the first Quora dataset, released by the end of January 2017.² It consists of over 400k pairs of questions labeled 1 in case they are duplicates of each other, or 0 otherwise. The pairs in the dataset released were collected with sampling techniques and their labeling may not be fully correct, and are not restricted to any subject (Iyer et al., 2017).

The other dataset used here is similar to one of the datasets used by Bogdanova et al. (2015). It is made of queries from the AskUbuntu forum,³ which are thus on a specific domain, namely from the IT area, in particular about the Ubuntu operative system. We used AskUbuntu dump available, from September 2014,⁴ containing 167,765 questions, of which 17,115 were labeled as a duplicate.

A portion with 30k randomly selected pairs of title+body was extracted, the same size as the portion used by Bogdanova et al. (2015). This portion is balanced, thus with an identical number of duplicate and non-duplicate pairs. To support the experiments described below, it was divided into 24k/6k for training/testing, an 80%/20% split.

²<https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs>.

³<https://askubuntu.com/>

⁴<https://meta.stackexchange.com/a/224922>.

The textual segments in this dataset contain both the title and the body of the query in the corresponding thread, and this dataset is referred to as AskUbuntuTB, while its counterpart with titles only—obtained by removing the bodies—is referred to as AskUbuntuTO.

To support comparison, a portion with 30k randomly selected pairs was extracted also from the Quora release, with the same duplicate vs. non-duplicate balance and the same training vs. test split rates as for the AskUbuntu dataset.

The average length of the segments in number of words is 84 in AskUbuntuTB. Its counterpart AskUbuntuTO, with titles only, represent a very substantial (10 times) drop to 8 words per segment on average, which is similar to the 10 words per segment in the Quora dataset.

The vocabularies sizes of AskUbuntuTB, AskUbuntuTO and Quora are 45k, 16k and 24k items, respectively, and their volumes are 5M, 500k and 650k tokens, respectively. Concerning the 400k pair Quora release, in turn, it contains 9M tokens and a 125k item vocabulary.

3 Rule-based

As a first approach experimented with, inspired by (Wu et al., 2011), we resorted to the Jaccard Coefficient over n -grams with n ranging 1 to 4.

Before applying this technique, the textual segments were preprocessed by undertaking (i) tokenization and stemming, using the NLTK tokenizer and Porter stemmer (Bird, 2006); and (ii) markup cleaning, whereby markup tags for references, links, snippets of code, etc. were removed.

To find the best threshold, we used the training set in a series of trials and applied the best results for the test sets. This led to the thresholds 0.1, 0.016, 0.03 for Quora, AskUbuntuTO and AskUbuntuTB, respectively.

This approach obtains 72.91% accuracy when applied over AskUbuntuTB.⁵

When running over AskUbuntuTO, its performance seems not to be dramatically affected by the much shorter segment length, suffering a slight decrease to 72.35%. Interestingly, a clear drop of the accuracy of over 3 percentage points is observed when it is run over Quora, scoring 69.53%.

These results seem to indicate that while this technique is quite robust with respect to the short-

⁵This is in line with the accuracy score of 72.65% reported by Bogdanova et al. (2015) with similar settings.

Title: vsftpd not listing large directory from WAN interface
Body: I have vsftpd running on my Ubuntu server, which is behind an Asus RT-N66U router. Port 21 is forwarded to the server. I can connect via my public IP address to the server (81 more words omitted)
Title: hiphop, nginx, spdy
Body: I'm about a month young at linux and brand new to ubuntu. I can do this to install hiphop https://github.com/facebook/hiphop-php/wiki/Building-and-installing-HHVM-on-Ubuntu-12.04 (69 more words omitted)
Title: No wireless ubuntu13.10
Body: Installed ubuntu 13.10 yesterday no internet connection.12.10 ok and dongle ok,13.4 no dongle, nothing now. Compac mini 110c, broadcom 4313 (AR8132 Q.Atheros.) Only have ubuntu on notebook.

Figure 3: Three example segments (titles and bodies) from AskUbuntu dataset

ening of the length of the segments, it is less robust when its application changes from a specific to an unconstrained domain.

4 Classifier

4.1 Basic features

To set up a DQD system resorting to an approach based on a supervised machine learning classifier, we resorted to supporting vector machines (SVM), following its acknowledged good performance in this sort of tasks and following an option also taken by Bogdanova et al. (2015). We employed SVC (Support Vector Classification) implementation from the sklearn support vector machine toolkit (Pedregosa et al., 2011)

For the first version of the classifier, a basic feature set (FS) was adopted. N -grams, with n from 1 to 4, were extracted from the training set and the ones with at least 10 occurrences⁶ were selected to support the FS

For each textual segment in a pair, a vector of size k was generated, where k is the number of n -grams included in the FS . Each vector encodes the occurrences of the n -grams in the corresponding segment, where vector position i will be 1 if the i -th n -gram occurs in the segment, and 0 otherwise. Then a feature vector of size $2k$ is created by concatenating the vectors of the two segments. This vector is further extended with the scores of the Jaccard coefficient determined over 1, 2, 3 and 4-grams. Hence, the final feature vector representing the pair to the classifier has the length $2k + 4$.

This system achieves 70.25% accuracy⁷ when trained over the AskUbuntuTB. Its accuracy drops some 1.5 percentage points, to 68.88%, when trained with the shorter segments of AskUbuntuTO, and drops over 5 points, to 64.93%, when

⁶We tried thresholds ranging from 5 to 15.

⁷We tried also with another implementation of SVM, namely SVM-light (Joachims, 2006), and the same score 70.25 was achieved.

trained with Quora, also with shorter segments than AskUbuntuTB but from a broader, all-encompassing domain.

4.2 Advanced features

To have an insight on how strong an SVM-based DQD resolver resorting to a basic FS like the one described above may be, we proceeded with further experiments, by adding more advanced features. We used Princeton WordNet (Fellbaum, 1998) to bring semantic knowledge to the system and used further text preprocessing to have more explicit lexical information, namely the text was normalized, e.g. “n’t” was replaced with “not”, etc., and POS tagged, with NLTK.

Lexical features The vector of each segment was extended with an extra feature, namely the number of negative words (e.g. *nothing*, *never*, etc.) occurring in it. And, to the concatenation of segment vectors, one further feature was added, the number of nouns that are common to both segments, provided they are not already included in the FS . Any pair was then represented by a vector of size $2(k + 1) + 4 + 1$.

Semantic features Eventually, any pair was represented by a vector of size $2(k + 1) + 4 + 2$, with its length being extended with yet an extra feature, namely the value of the cosine similarity between the embeddings of the segments in the pair.

For a given segment, its embedding, or distributional semantic vector, was obtained by summing up the embeddings of the nouns and verbs occurring in it, as these showed to support the best performance after experiments have been undertaken with all parts-of-speech and their subsets.

The embeddings were based on WordNet synsets, rather than on words, as these were shown to lead to better results after experimenting with both options. We employed word2vec word embeddings (Mikolov et al., 2013) and used Autoex-

tend (Rothe and Schütze, 2015) to extract synset embeddings with the support of WordNet. We adopted the same configuration as in that paper and used version 3 of WordNet, which contains over 120k concepts, represented by synsets. The main advantage of synset embeddings over word embeddings in duplicate detection is the fact that synonyms receive exactly the same distributional vectors, which helps to appropriately take into account words in the segments of the pair that are different in linguistic form but are synonyms.

Results The resulting system permitted an improvement of over 5 percentage points with respect to its previous version trained with basic features, scoring 75.87% accuracy when running over AskUbuntuTB.

This advantage is not so large when it is run over the datasets with shorter segments. It scored 70.87% with AskUbuntuTO (positive delta of almost 2 points relative to the previous basic version), and 68.56% with Quora (over 3.5 points better).⁸

5 Neural Networks

We experimented with three different architectures for DQD resolvers based on neural networks. The first experiment adopts the architecture explored in one of the papers reporting the most competitive results for DQD, and the second adopts the neural architecture of the top performing system in the “Question-Question” subtask of SemEval2016. The third system adopts a hybrid architecture combining key ingredients of the previous two.

5.1 Convolutional

The architecture of convolutional neural network (CNN) to address DQD was introduced by Bogdanova et al. (2015). First, the CNN obtains the vectorial representations of the words, also known as word embeddings, in the two input segments. Next, a convolutional layer constructs a vectorial representation for each one of the two segments. Finally, the two representations are compared using cosine similarity, whose value if above an empirically estimated threshold, determines that the two segments are duplicate (diagram in Figure 4).

⁸This score was obtained resorting to 1- to 4-grams. Experiments with 1- to 3-grams and with 1- to 5-grams delivered worst scores, respectively 68.38% and 68.42%.

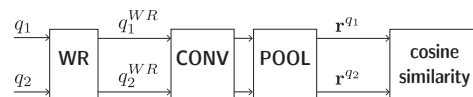


Figure 4: CNN architecture: word representation (WR), convolution (CONV), pooling (POOL) and cosine similarity measurement layers.

To replicate this approach, we resorted to Keras (Chollet, 2015) with Tensorflow (Abadi et al., 2015) back-end for training and evaluating the neural network. The hyper-parameters either replicate the ones reported by Bogdanova et al. (2015) or are taken from vanilla CNN architecture as it is implemented in the above libraries.

The DeepLearning4j⁹ toolkit was used for creating the initial word representations. Bogdanova et al. (2015) specify only the skip-gram neural network architecture and the embeddings dimensionality of 200 as training parameters for their best run. In our experiment, besides these parameters, all the other hyper-parameters were taken from a vanilla version of *word2vec* implemented in DeepLearning4j. In this experiment, to train word embeddings, we used the 38 million tokens of the September 2014 AskUbuntu dump available.¹⁰

When trained over AskUbuntuTB, the system performs with 73.40% accuracy. An improvement of over 1 point, to 74.50%, was obtained with a slight variant where the CNN was run without pre-trained word embeddings, and with a random initialization of the embeddings using uniform distribution.

The drop in performance observed in the systems presented above when moving to shorter segments is also observed here, with a much greater impact with Quora, coming down almost 15 points, to 59.90%, than with AskUbuntuTO, which comes down less than half a point, to 74.10%. This seems to indicate that the CNN is less robust than previous approaches when moving from a specific to a generic domain.

The score of 73.40%, obtained with settings similar to Bogdanova et al. (2015), is inferior in almost 20 percentage points to the score reported in that paper. This led us to look more carefully in the two experiments.

As indicated in previous sections, in our experi-

⁹<http://deeplearning4j.org>

¹⁰Bogdanova et al. (2015) used 121 million tokens from the May 2014 dump available to them.

ments the datasets were submitted to a preprocessing phase, including markup cleaning by means of which tags for references, links, snippets of code, etc. were removed. One of these tags is rendered to the reader of a thread in the AskUbuntu forum as “Possible duplicate: <title>”, where <title> is instantiated with the title of the other thread that the present one is a possible duplicate of, and is linked to the page containing that other thread.

As we hypothesized that this might be a reason for the 20 point delta observed, we retrained our CNN-based system over AskUbuntuTB slightly modified just to keep that “Possible duplicate <title>” phrase. Accuracy of 94.20% was obtained, in the same range of the 92.9% score reported by Bogdanova et al. (2015).¹¹

5.2 Deep

MayoNLP (Afzal et al., 2016) was the top performing system in the “Question-Question” subtask of SemEval 2016 Task 1 (Agirre et al., 2016).

Its architecture is based on Deep Structured Semantic Models, introduced by Huang et al. (2013), whose first layer is a 30k dense neural network followed by two hidden multi-layers with 300 neurons each and finally a 128 neuron output layer. All the layers are feed-forward and fully connected (diagram in Figure 5).

This neural network was used to process text and given the huge dimension of the input text (around 500k tokens), a word hashing method was used that creates trigrams for every word in the input sentence: for instance, the word *girl* would be represented as the trigrams *#gi*, *gir*, *irl* and *rl#*, including the beginning and end marks. This permitted to reduce the dimension of the input text to 30k, which is represented in the first neural layer.

The MayoNLP system adopts this architecture with the difference that the two hidden layer become a 1k neuron layer and the output layer is adapted to the SemEval2016 subtask, which is a graded textual similarity classification.

We resorted to the Keras deep learning library to replicate this architecture. Given that the dimension of the input in our task was smaller, we used one neuron for each word in our vocabulary and it was not necessary to resort to word hashing for dimensionality reduction. Hence, an input layer with approximately the same size of neurons was

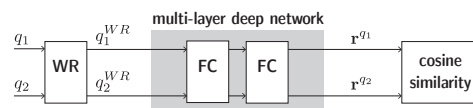


Figure 5: DNN architecture: word representation layer (WR), fully connected layers (FC) and cosine similarity measurement layer.

created: 63k for the AskUbuntuTB dataset, 16k neurons for AskUbuntuTO and 24k for Quora.

When evaluating the resulting system, the same overall pattern as with previous approaches emerges. The best accuracy is obtained with AskUbuntuTB, 78.65%, which has a slight drop with AskUbuntuTO, to 78.40%.

These scores are in contrast with the accuracy of 69.53% obtained with Quora, indicating that also here moving to a generic domain imposes a substantial loss of accuracy, of over 8 points.¹²

5.3 Hybrid DCNN

We also experimented with a novel architecture we developed by combining the convoluted and deep models discussed in the previous sections. By resorting to the Keras deep learning library, the key ingredients of the convoluted and the deep networks (DCNN) were implemented together.

The hybrid DCNN starts with the same input structure as the CNN, obtaining the vectorial representations of words in two input segments. It then connects each of them to a shared convolutional layer followed by three hidden and fully connected layers, whose output is finally compared using the cosine distance. Both the convolutional and the deep layers share the same weights for the two sentences input, in a siamese network (diagram in Figure 6).

The vectorial representation uses an embedding layer of 300 randomly initiated neurons with uniform distribution which are trainable. The convolution layer uses 300 neurons for the output of filters with a kernel size of 15 units, and each deep layer has 50 neurons.

Differently, from previous approaches, the resulting DQD resolver scores better over AskUbuntuTO, scoring 79.67%, than over AskUbuntuTB, for which it gets a 79.00% accuracy score. This

¹¹Our attempt to reach the authors to obtain a copy of the dataset used in their paper remained unreplied.

¹²In the “Question-Question” subtask of SemEval 2016, thus with different datasets and for the different task of 0 to 5 graded similarity classification task, the MayoNLP system scored 0.73035 in terms of Pearson correlation coefficient (Agirre et al., 2016).

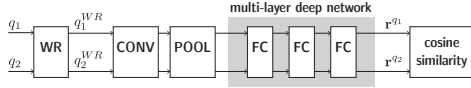


Figure 6: DCNN architecture.

may be an indicator that, when using the title and body, the neural network could perform better but may be failing due to the sparseness of the data, which requires possibly a higher number of neurons in the deep layer.

As for the result with Quora, in turn, the same pattern is observed as in previous systems. There is a substantial drop of over 8 points, to 71.48%.

6 Discussion

The experimental results reported in the previous sections are summarized in Table 1. The performance of each approach or architecture for DQD was assessed in the respective section. Putting all results side by side, some patterns emerge.

Shortening the length of the segments (from 84 words per segment on average, with AskUbuntuTB, to 8 or 10 words, respectively with AskUbuntuTO or Quora) has an overall negative impact on the accuracy of the systems, except for DCNN. For AskUbuntuTO, the negative delta ranges from 0.25 points, with DNN, to over 5 points, with SVM-adv.

NN-based solutions seem thus to be more robust to the shortening of the length of the segments than SVM-based ones, even to the point where the more sophisticated DCNN approach inverts this pattern, and performs better for shorter segments than for longer ones with AskUbuntu.

As the average length of segments in AskUbuntuTO and Quora are similar, the contrast between their scores permits to uncover yet another pattern. Moving from a specific to a generic domain has an overall negative impact on the accuracy of the systems, which is wider than with the shortening of the segments. The negative delta ranges from less than 3 points, with Jaccard or SVM-base, to over 14 points, with DNN.

The level of the impact seems to be inverted here. It is the non NN-based solutions that appear as more robust to the generalization of the domain than the NN-based ones, to the point that the superiority shown by NN-based ones with the specific domain is reduced or even canceled with the general domain.

It is interesting to note that, for the generic domain, the CNN approach offers the worst result. The DNN overcomes the best SVM approach by less than 2 points. And only the DCNN overcomes the overall second-best, but also by a modest margin.

It is also very interesting to note that, for general domain, the rule-based approach is one of the two second best, thus challenging the immense sophistication of any other approach, including the NN-based ones.

7 Cross-domain application

Given the scarcity of labeled datasets of pairs of interrogative segments, in real usage scenarios systems tend to be trained on as much data as possible from all sources and different domains. We experimentally approximated this scenario by training the best DQD system of each approach over the generic dataset (Quora) and evaluating them over the focused dataset (AskUbuntuTO).

The rule-based, the advanced SVM and the DCNN perform, respectively, with accuracies of 57.63% (dropping almost 15 points), 53.50% (dropping over 17 points) and 56.42% (dropping over 15 points).

Interestingly, leaner systems seem to be more robust in this approximation to its application in real-usage scenarios than more sophisticated ones. Importantly, however, for all types of systems, accuracy is observed to degrade and comes close to random decision performance.¹³

Given these results, it is interesting to check, for the generic domain, how this challenge may evolve when the training set is substantially enlarged, namely to the largest dataset available at present, that is to the over 400K pairs of the full Quora release. We picked the more lean technique (rule-based) and more sophisticated one (DCNN).

Interestingly their accuracy scores evolved towards opposite directions when compared to the scores obtained with the (over 13 times) smaller experimental Quora dataset, with 30k pairs. The rule-based solution scored 66.74% accuracy, dropping over 2 points, while the DCNN scored 79.36%, progressing almost 8 points.¹⁴

¹³Recall that in our experimental conditions, the 30k datasets are balanced with 50% duplicate pairs and 50% non-duplicate, but the 400k Quora release is not. To confirm this trend, we collected yet another data point for CNN, which scored 50.20%, in line with the other systems.

¹⁴Given the huge volume of the Quora release, we imple-

	Jcrd	SVM-bas	SVM-adv	CNN	DNN	DCNN
AskUbuntu						
title and body	72.91	70.25	75.87	74.50	78.65	79.00
title only	72.35	68.88	70.87	74.12	78.40	79.67
Quora	69.53	64.93	68.56	59.90	69.53	71.48

Table 1: Accuracy of the 6 systems (columns) over the 3 datasets (lines)

8 Related work

An interesting approach to DQD was introduced by Wu et al. (2011). It resorts to the Jaccard coefficient to measure similarities between two segments in the pair. Separate coefficients are calculated, and assigned different weights, for the segments. A threshold is empirically estimated and used to determine whether two threads are duplicates. An f-score of 60.29 is obtained for the titles only, trained with 3M questions and tested against 2k pairs taken from a dataset obtained from Baidu Zhidao, in Chinese. This approach is used as a baseline by Bogdanova et al. (2015). This system inspired one of the architectures used in our experiments, presented in detail in Section 5.1.

The recent SemEval-2016 Task 1 included a “Question-Question” subtask to determine the degree of similarity between two interrogative segments. The MayoNLP system (Afzal et al., 2016) obtained the best accuracy in this task. This system inspired one of the systems used in our experiments, presented in detail in Section 5.2.

Regarding the Quora dataset released 2 months ago, to the best of our knowledge, up to now there is only one unpublished paper concerning that task (Wang et al., 2017). It proposes a multi-perspective matching (BiMPM) model and evaluates it upon a 96%/2%/2% train/dev/test split. This system is reported to reach an accuracy of 88.17%.

Other draft results concerning Quora dataset are available only as blog posts^{15,16} and are based on the model for natural language inference proposed by Parikh et al. (2016).

mented a lean version of DCNN to run this experiment, which used a vectorial representation of 25 neurons randomly initiated, followed by a convolution layer which uses 10 neurons for the output of filters and with a 5 kernel size. The deep layers were reduced to two layers each with 10 neurons. A 70%/30% randomly extracted for training/testing was used both for the experiment with the DCNN and with the rule-based approach.

¹⁵<https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning>

¹⁶<https://explosion.ai/blog/quora-deep-text-pair-classification>

9 Conclusions

The experiments reported in this paper permitted to advance the understanding of the duplicate question detection task and improve its application. There is consistent progress in terms of the accuracy of the systems as one moves from less to more sophisticated approaches, from rule-based to support vector machines, and from these to neural networks, when its application is over a narrow, specific domain. The same trend is observed for the range of support vector machines solutions, with better results obtained for resolvers resorting to more advanced features. And it is observed also for the range of neural network architectures experimented with, from convoluted to deep networks, and from these to hybrid convoluted deep ones. *Overall, the novel neural network architecture we propose presents the best performance of all resolvers tested.*

The rate of this progress is however mitigated or even gets close to be canceled when one moves from a narrow and specific to broad and all-encompassing domain. Under our experimental conditions, the gap of over 11 points from the worst to the best performing solution with a narrow domain is cut to almost half, and the more sophisticated solution, with the best score, overcomes the leanest one just by less than 2 points when running over a generic domain.

Interestingly, when one moves, in turn, from longer to (eight times) shorter segments, only minor drops in performance are registered.

Given the scarcity of labeled datasets of pairs of interrogative segments, in real usage scenarios, systems are trained on as much data as possible from all sources and different domains and eventually applied over narrow domains. We experimentally approximated this scenario, where the accuracy of the systems was observed to degrade and come close to random decision performance.

In future work, we will extend our experimental space to further systems and conditions, including larger datasets, and languages other than English.

Acknowledgements

The present research was partly supported by the CLARIN and ANI/3279/2016 grants.

References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. [TensorFlow: Large-scale machine learning on heterogeneous systems](https://www.tensorflow.org/). Software available from [tensorflow.org](https://www.tensorflow.org/). [https://tensorflow.org/](https://www.tensorflow.org/).
- Naveed Afzal, Yanshan Wang, and Hongfang Liu. 2016. MayoNLP at SemEval-2016 task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1258–1263.
- Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. [Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation](https://www.aclweb.org/anthology/S/S16/S16-1081.pdf). In Steven Bethard, Daniel M. Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16–17, 2016*. The Association for Computer Linguistics, pages 497–511. [http://aclweb.org/anthology/S/S16/S16-1081.pdf](https://www.aclweb.org/anthology/S/S16/S16-1081.pdf).
- Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, pages 69–72.
- Dasha Bogdanova, Cícero Nogueira dos Santos, Luciano Barbosa, and Bianca Zadrozny. 2015. [Detecting semantically equivalent questions in online user forums](https://www.aclweb.org/anthology/K/K15/K15-1013.pdf). In Afra Alishahi and Alessandro Moschitti, editors, *Proceedings of the 19th Conference on Computational Natural Language Learning, CoNLL 2015, Beijing, China, July 30–31, 2015*. ACL, pages 123–131. [http://aclweb.org/anthology/K/K15/K15-1013.pdf](https://www.aclweb.org/anthology/K/K15/K15-1013.pdf).
- François Chollet. 2015. [Keras](https://github.com/fchollet/keras). <https://github.com/fchollet/keras>.
- Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. ACM, pages 2333–2338.
- Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. [First quora dataset release: Question pairs](https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs). <https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs>.
- Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. [Linguistic regularities in continuous space word representations](https://www.aclweb.org/anthology/N/N13/N13-1090.pdf). In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9–14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. The Association for Computational Linguistics, pages 746–751. [http://aclweb.org/anthology/N/N13/N13-1090.pdf](https://www.aclweb.org/anthology/N/N13/N13-1090.pdf).
- Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. [A decomposable attention model for natural language inference](https://www.aclweb.org/anthology/D/D16/D16-1244.pdf). In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016*. The Association for Computational Linguistics, pages 2249–2255. [http://aclweb.org/anthology/D/D16/D16-1244.pdf](https://www.aclweb.org/anthology/D/D16/D16-1244.pdf).
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.
- Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*.
- Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. [Bilateral multi-perspective matching for natural language sentences](https://arxiv.org/abs/1702.03814). *CoRR* abs/1702.03814. [http://arxiv.org/abs/1702.03814](https://arxiv.org/abs/1702.03814).
- Yan Wu, Qi Zhang, and Xuanjing Huang. 2011. [Efficient near-duplicate detection for Q&A forum](https://www.aclweb.org/anthology/I/I11/I11-1112.pdf). In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8–13, 2011*. The Association for Computer Linguistics, pages 1001–1009. [http://aclweb.org/anthology/I/I11/I11-1112.pdf](https://www.aclweb.org/anthology/I/I11/I11-1112.pdf).