Boosting a rule-based chatbot using statistics and user satisfaction ratings

Octavia Efraim, Vladislav Maraev, João Rodrigues

Boosting a rule-based chatbot using statistics and user satisfaction ratings

Octavia Efraim
Univ Rennes, LIDILE EA 3874, France
octavia-edie.efraim@univ-rennes2.fr
Vladislav Maraev
CLASP, University of Gothenburg, Sweden
vladislav.maraev@gu.se
João Rodrigues
Department of Informatics, Faculty of Sciences, University of Lisbon, Portugal
joao.rodrigues@di.fc.ul.pt

The final publication is available at Springer via https://doi.org/10.1007/978-3-319-71746-3_3.

Abstract. Using data from user-chatbot conversations where users have rated the answers as good or bad, we propose a more efficient alternative to a chatbot's keyword-based answer retrieval heuristic. We test two neural network approaches to the near-duplicate question detection task as a first step towards a better answer retrieval method. A convolutional neural network architecture gives promising results on this difficult task.

1 Introduction

A task-oriented conversational agent which returns predefined answers from a fixed set (as opposed to generating responses in real time) can provide a considerable edge over a fully-human answering system, if it handles correctly most of the repetitive queries which require no personalised answer. Indeed, at least in our experience, many of the questions asked by users and their expected answer look like entries in a list of frequently asked questions (FAQ): "What are your opening hours?", "Do you deliver to this area?", etc. An effective conversational agent, or chatbot, can act as a filter, sifting out such questions and only passing on to human agents those it is unable to deal with: those which are too complex (e.g. made up of multiple queries), those for which there simply is no response available, or those which require consulting a client database in order to provide a personalised answer (e.g. the status of a specific order or request). Such questions may occur at the very beginning or at some later point during a conversation between a customer and the automated agent. In the latter case, a well-performing chatbot will at least have saved human effort up to the moment where the difficulty emerged (provided it also hands on to the human a summary of the dialogue).

If the job of such retrieval-based conversational agents may seem easy enough to be successfully handled through a rule-based approach, in reality, questions coming from users exhibit much more variation (be it lexical, spelling-related, or syntactic)

that is feasibly built into hand-crafted rules for question parsing. Approaches based on statistical learning from data may therefore benefit such answer retrieval systems.

Our goal is to improve on an existing closed-domain chatbot which returns answers from a closed set using keywords as a retrieval heuristic and human-defined priority rules to break ties between multiple candidate answers. Assuming a question does have an answer in the closed answer repository, this chatbot may fail to find it because it misunderstands the question (in which case it replies with the wrong answer) or because it is unable to "understand" it (i.e. map it to an available response) altogether (it then asks the user to provide an alternative formulation). This design means that the chatbot's ability to recognise that two distinct questions can be accurately answered by the same reply is very limited. Potential improvements to this system design may target the answer retrieval method, the candidate answer ranking method, and the detection of out-of-domain questions. We choose to address answer retrieval.

This paper is organised as follows: in Section 2 we review some tasks and solutions which are potentially relevant to our goal; Section 3 gives an overview of the system we set out to improve; Section 4 describes the data available to us, and our problem formulation; in Section 5 we outline the procedure we applied to our data in order to derive from it a new dataset suited to our chosen task; Section 6 gives an account of our proposed systems; in Section 7 we sum up and discuss our results; finally, Section 8 outlines some directions for follow-up work.

2 Related work

The ability to predict a candidate answer's fitness to a question is a potentially useful feature in a dialogue system's answer selection module. A low-confidence score for a candidate answer amounts to a problematic turn in a conversation, one that warrants corrective action. Addressing **success/failure prediction in dialogue**, both [28] (human-computer dialogues in the customer relationship domain) and [23] (human-human task-oriented dialogues) distinguish between a predictive task with immediate utility for corrective action in real time, and a post-hoc estimation task for analysis purposes. If the former authors learn a set of classification rules from meta-textual and meta-conversational features only, the latter find that, with an SVM classifier, lexical and syntactic repetition reliably predict the success of a task solved via dialogue.

Answer selection for question answering has recently been addressed using deep learning techniques. In [8], for instance, the task is treated as a binary classification problem over question-answer (QA) pairs: the matching is appropriate or not. The authors propose a language-independent framework based on convolutional neural networks (CNN). The power of 1-dimensional (1D) convolutional-and-pooling architectures in handling language data stems from their sensitivity to local ordering information, which turns them into powerful detectors of informative n-grams [9]. Some of the CNN architectures and similarity metrics tested in [8] on a dataset from the insurance domain achieve good accuracy in selecting one answer from a closed pool of candidates.

The answer selection problem has also been formulated in terms of **information retrieval**. For example, [15] reports on an attempt to answer open-domain questions asked by users on Web forums, by searching the answer in a large but limited set of FAQ QA

pairs collected in a previous step. The authors use simple vector-space retrieval models over the user's question treated as a query and the FAQ question, answer, and source document indexed as fields making up the item to be returned. Also taking advantage of the multi-field structure of answers in QA archives, [31] combines a translation-based language model estimated on QA pairs viewed as a parallel corpus, and a query likelihood model with the question field, the answer field, and both combined. A special application of information retrieval, **SMS-based FAQ retrieval** – which was proposed as a shared task at the Forum for Information Retrieval Evaluation in 2011 and 2012 – faces the additional challenge of very short and noisy questions. The authors of [11] break the task down into: question normalisation using rules learnt on several corpora annotated with error corrections; retrieval of a ranked list of answers using a combination of a term overlap metric and two search engines with BM25 as the ranking function, over three indexes (FAQ question, FAQ answer, and both combined); finally, filtering out-of-domain questions using methods specific to each retrieval solution.

Equating new questions to past ones that have already been successfully answered has been proposed as another way of tackling question answering. Such **duplicate question detection** (DQD) approaches fall under near-duplicate detection, and are related to paraphrase identification and other such instances of the broader problem of textual semantic similarity, with particular applications, among others, to community question answering (*cf.* Task 3 at SemEval-2015, 2016, and 2017). In turn, DQD may be cast as an information retrieval problem [4], where the comparison for matching is performed on different entities: new question with or without its detailed explanation if available, old question with or without the answer associated with it; where the task is not to reply to new questions, but rather to organise a QA set, answers have even been compared to each other in order to infer the similarity of their respective questions [14]. Identifying semantically similar questions entails at least two major difficulties: similarity measures targeted at longer documents are not suited to short texts such as regular questions; and word overlap measures (such as Dice's coefficient or the Jaccard similarity coefficient) cannot account for questions which mean the same but use different words. Notwithstanding, word overlap features have been shown to be efficient in certain settings [22, 13]. CNN architectures, which, since their adoption from computer vision, have proved to be very successful feature extractors in text processing [9], have recently started to be applied to the task of DQD. [6] reports impressive results with word-based CNN on data from the StackExchange QA forum. In [25], the authors obtain very good performance on a subset of the AskUbuntu section of StackExchange by combining a similar word-based CNN with an architecture based on [2].

Answer relevancy judgements by human annotators on the output of dialogue systems are a common way of evaluating this technology. The definition of relevancy is tailored to each experimental setup and research goal. In [24] annotators assess whether the answer generated by a system based on statistical machine translation in reply to a Twitter status post is on the same topic as that post and "makes sense" in response to it. More recently—to cite just one example taken from a large body of work on neural response generation—, to evaluate the performance of the neural conversation model in [27], human judges are asked to choose the better of two replies to a given question: the output of the experimental system and a chatbot. The role of human judgements in

such settings is nonetheless purely evaluative: the judge assesses post hoc the quality of a small sample of the system output according to some relevancy criterion. In contrast to these experiments, ours is not an unsupervised response generation system, but a **supervised retrieval-based system**, as defined in [19], insofar as it does "explicitly incorporate some supervised signal such as task completion or user satisfaction". Our goal is to take advantage of this feature not only for evaluation, but also for the system's actual design. As far as the evaluation of unsupervised response generation systems goes, this is a challenging area of research in its own right [19, 18].

3 Overview of the rule-based chatbot

The chatbot we are aiming at improving is deployed on the website of a French air carrier as a chat interface with an animated avatar. The system was developed by a private company and we had no participation in its conception or implementation. Its purpose is, given a question, to return a suitable predefined answer from a closed set. The French-speaking chatbot has access to a database of 310 responses, each of which is associated unambiguously with one or more keywords and/or skip-keyphrases (phrases which allow for intervening words). An answer is triggered whenever the agent detects in the user's query one of the keywords or keyphrases associated with that answer. A set of generic priority rules is used to break ties between competing candidate answers (which are simultaneously induced by the concurrent presence in the question of their respective keywords).

While this chatbot is closed-domain (air travel), a few responses have been included to handle general conversation (weather, personal questions related to the chatbot, etc.), usually prompting the user to go back on topic. A few other answers are given in default of keywords in the query: the chatbot informs the user that it has not understood the question, and prompts them to rephrase it. Some answers include one or several links either to pages on the company's website or to another answer; in the latter case, a click on the link will trigger a pseudo-question (a query is generated automatically upon the click, and recorded as a new question from the user). By virtue of its design, this system is deterministic: it will always provide the same answer given the same question.

The user interface provides a simple evaluation feature: two buttons (a smiling face and a sad face) enabling users to mark an answer as relevant or irrelevant to the query that prompted it. This evaluation feature is optional and not systematically used by customers. Exchanges with the chatbot usually consist of a single QA pair. There are, however, longer conversations too. Such dialogues can span a few minutes up to many hours, as no limit is imposed on the duration of a period of inactivity (the dialogue box does not close automatically). We improperly denote all input coming from a user as a question: in fact, in longer conversations a message can be phatic, evaluative of the previous answer of the chatbot's performance, it may convey information, or it may be asking a question properly.

4 Raw data and task definition

4.1 Data

Our original data consists of QA pairs from conversations with the chatbot, where users have rated the system's answer using the smiley button. For our purposes, a smiling face rating amounts to a label of "good" and a sad face rating to a label of "bad", assigned to the answer in relation to the question. This binary assessment scheme is far from the complexity of the many multidimensional evaluation frameworks that have been proposed over time to assess the subjective satisfaction or acceptance of users of dialogue systems, chiefly spoken ones [29, 12]. But, while a more nuanced evaluation scale might have been desirable, this simple binary scheme (which is not of our making, but was built into the system) is also lighter on the user. We do not equate the binary judgements with an objective measure of task success, because of their subjective component: many aspects of the user's experience with the system may influence the rating. Therefore we term the "good/bad" ratings in our data "user satisfaction ratings".

We have limited ourselves to one-turn dialogues (which are also the most common), in order to deal with self-contained questions. Our dataset contains 48,114 QA pairs from one-line dialogues. The proportions of classes are 0.28 for "good" and 0.72 for "bad". We conjecture that the predominance of negative ratings is partly a matter of negativity bias [3]: since customers are free, but not required, to evaluate the chatbot's answer, they may choose to do so mostly when they have strong feelings (which are more often negative) about it. Questions are relatively short (13 words and 70 characters on average; median values: 11 words and 57 characters), but there are a few outliers (a maximum of 241 words and 1357 characters).

4.2 Approach chosen

As mentioned above (Section 1), our goal is to improve the chatbot's performance on retrieving answers. We break down the answer retrieval problem into two steps:

1. **Duplicate question detection (DQD)**. Given a question, classify it as semantically similar or dissimilar to questions from a set of past questions with known answers.
2. **Answer selection**. Select an answer to the new question based on the DQD output.

In this paper we address the DQD task. We **define semantic similarity** for the task at hand in line with the definition of semantic equivalence in [6], with an additional requirement as per [22]: **we take two questions to be semantically similar if they can be correctly answered by the same answer, whose hypothetical existence suffices, provided that this answer is as specific as possible**. As a point of terminology, "similarity" seems more permissive than "equivalence" as to how far two questions are allowed to diverge from one another: "What time does the flight to New York depart on Monday 12th?" and "When is the departure time for NY on Monday 26th?" may be considered similar because they instantiate the same underlying question ("What is the departure time for New York on Mondays?"), but not strictly equivalent, since the actual details (the dates) differ.

A successful approach to answer retrieval based on DQD addresses our desired improvements to the rule-based chatbot system. It improves the retrieval performance, since it results in more questions being successfully linked to their correct answer. Additionally, the tool can present the user with a set of candidate answers if it is not confident enough to select one.

5 Data preparation

From the original set of question-answer-label triplets, we produced **a set of question-question (QQ) pairs labelled for semantic similarity**. The transformation we applied to the data is equivalent to interpreting the result of the chatbot's retrieval heuristic in terms of DQD. Thus, all questions answered correctly by a particular answer make up a set of semantically similar questions; all questions answered incorrectly by a particular answer form pairs of semantically dissimilar questions with each of the questions for which that same answer is correct. In line with this interpretation, we generated QQ pairs as described below.

First, we grouped all the questions in our dataset by the answer they received. At this point, each answer is linked to a set of questions for which users have rated it as a good answer (its "positive" group), and to another set of questions for which it has been rated as bad (the answer's "negative" group). Second, we selected a subset of most rated (either as good or as bad) and most informative answers. We discarded very generic answers (e.g. greetings, thanks) and those stating the chatbot's inability to understand the question. An analysis of the distribution of answers in the dataset then revealed that, of the remaining 246 unique answers, the 40 answers with the highest number of total "positive" and "negative" questions made up 79% of the dataset overall, 73% of all "positive" questions, and 81% of all "negative" questions. Those 40 answers were the ones we selected for learning, since they are arguably the most useful ones: they are the most frequently given ones overall, and also comprise both the best-rated answers and the most heavily rejected by users. Next, for each of the 40 answers, we generated exhaustively: pairs of "positive" questions – these are pairs of semantically similar questions (according to our definition of semantic similarity); and pairs made up of one "positive" and one "negative" question – these are pairs of semantically dissimilar questions. Lastly, to keep the data for learning of manageable size, we sampled QQ pairs from the full pairings generated at the previous step. In order to avoid issues related to learning from an imbalanced dataset (there are more dissimilar than similar pairs), we took an equal number of similar and dissimilar pairs, by randomly sampling 10,000 similar pairs and 10,000 dissimilar pairs, which amounts to undersampling the majority class.

6 Experimental setup

6.1 Data preprocessing

Questions in our dataset share many features with SMS and with other types of user-generated content, such as social media. The text is riddled with spelling mistakes (e.g.

merci mais ca ne me precise pas le retard de marseille la reunion le 25 09 a 190h et j essai de vous appeler en vains car au bout de 15 mn ca racroche), but also with the deliberate use of simplifying and expressive devices [26]: repeated punctuation, capitalisation, graphemic stretching, emoticons (e.g. *merciiiiiiii :) :)*, *NON NON NON!!!!!!!!! J'ai besoin du numero de vol de CDG a JFK qui arrive ce soir*).

For our task, the text of the questions in our QQ dataset underwent a number of cleaning and preprocessing steps. We cleaned up HTML markup and entities, and certain characters. Basic normalisation included lowercasing, removing punctuation, collapsing sequences of more than two repeated characters [1], restoring elided vowels, standardising spelling variations of in-domain terms and proper names and merging those which are multi-word (e.g. *ny*, *nyc*, *new york*, and *newyork* all become *newyork*; *AR*, *aller-retour*, <>, etc. are all replaced by *allerretour*), and grouping sequences that match specific patterns under semantic and formal classes inspired from Bikel et al. [5]: dates, telephone numbers, prices, measurements, other numeric expressions, URLs, e-mail addresses, etc. Given the poor performance and strong disagreement of four language detection packages that we tried on our data, most probably due to the very short size of our questions, we abandoned the idea of automatically filtering out questions in a language other than French. We produced five versions of the text:

1. **Preprocessed** as described above.
2. **Lemmatised** using *MElt* [7] on the preprocessed text. *MElt* is a maximum-entropy Markov-model POS tagger and lemmatiser with a normaliser/corrector wrapper trained on user-generated corpora annotated by hand. Some post-lemmatisation cleaning was needed, mainly for lemma disambiguation.
3. **PoS**: a version of the preprocessed text where tokens were replaced with their part-of-speech tags as output by *MElt*.
4. **Stemmed** using Porter's algorithm on the lemmatised version.
5. **Stemmed after removing accents from lemmas**. Because customers use accents rather haphazardly, it seems reasonable to assume that reducing word forms to stems after stripping accents may decrease considerably the size of the vocabulary.

6.2 Baselines

Our weak baseline is the chatbot in its current form, taken as an (indirect) detector of similar questions. The construction procedure of our QQ dataset means that this baseline has 50% accuracy on our class-balanced data. Indeed, the chatbot correctly identifies all the similar QQ pairs as similar, but it also takes all the dissimilar ones for similar.

For the remainder of systems, including the second baseline, the same train/test split on the data was used, with an 80/20 ratio. We take as strong baseline the Jaccard similarity coefficient, a measure of overlap between sets which is common in information retrieval [21], and which has been used for textual entailment recognition [20] and for near-duplicate detection tasks [30]. For each QQ pair, we compute the Jaccard coefficient between the two questions represented as a set of n-grams (with n running from 1 to 4). The cutoff value is optimised on the training set, and evaluated on the test set.

6.3 Proposed systems

The systems we are testing are two CNN architectures developed specifically for DQD, which performed very well on a dataset in English from the AskUbuntu forum [6]. CNN architectures have shown great success at a number of natural language processing tasks, such as classifying sentences [16] or modelling sentence pairs [32].

CNN Our system is based on the CNN architecture for DQD proposed in [6]. First, the CNN obtains vector representations of the words, also known as word embeddings, from the two input segments. Next, a convolution layer constructs a vector representation for each of the two segments. Finally, the two representations are compared using cosine similarity. If the value of this metric exceeds an empirically estimated threshold, the two segments are classified as duplicate. The same feature maps (for word embedding and the convolution layer) are used to produce the representation of both questions.

Our CNN architecture is also inspired from [17]. The authors of that paper use the concatenation of several convolution filters with multiple feature widths. We improve the architecture proposed in [6] by changing the convolution layer to a set of convolution filters with multiple feature widths (*cf.* diagram in Figure 1).

The vector representation uses an embedding layer of 200 randomly initiated neurons which are trainable. Each convolution layer uses 100 neurons for the output of the filters, and the widths of the filters are 2, 3, and 5. The optimisation algorithm used for the network is stochastic gradient descent (SGD) with a learning rate of 0.005.

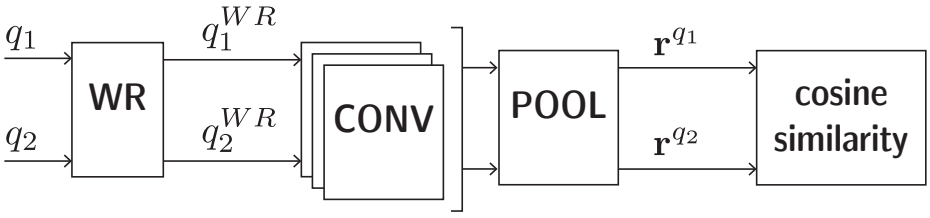

Fig. 1. CNN architecture, with layers: word representation (WR) for a pair of questions (q_n; output q_n^{WR}); concatenated set of convolution filters (CONV); max pooling (POOL); question representation ($\mathbf{r}^{q_n}$); and cosine similarity measurement.

Hybrid deep CNN (CNN-Deep) The second system we tested is described in detail in (selfReference). It combines a CNN similar to our first proposed system with a deep neural network with three hidden, fully-connected, layers, based on the architecture described in [2]. A diagram of the system is shown in Figure 2.

The vector representation uses an embedding layer of 300 randomly initiated neurons 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. The optimisation algorithm used for the network is SGD with a learning rate of 0.01.

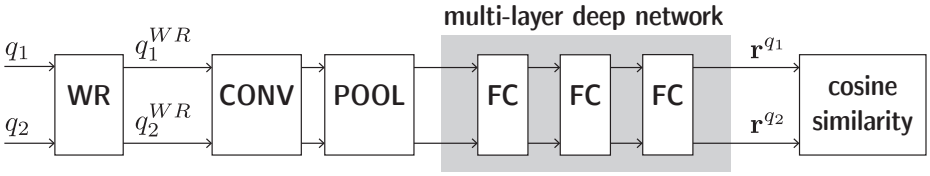

Fig. 2. CNN-Deep architecture: as the CNN, with the addition of fully-connected layers (FC).

7 Results and discussion

7.1 System performance

From Table 1, it is immediately obvious that two of the three systems (Jaccard and CNN-Deep) perform barely better than a random classifier (such as our weak baseline, which has accuracy of 50%), while CNN is at the order of 20 percentage points above both on four of the five text versions.

	Jaccard	*CNN*	*CNN-Deep*
Preprocessed	52.8	**74.9**	55.8
Lemmatised	55.2	72.5	53.0
PoS	51.6	59.7	55.7
Stemmed	54.3	72.4	56.0
Stemmed unaccented lemmas	54.2	72.1	55.3

Table 1. Accuracy for the DQD task on the five versions of the data.

The poor accuracy of the Jaccard similarity baseline goes to confirm that, for our task, word overlap is not a reliable indicator of semantic similarity. For example, the questions in pair 1 in Table 2 are similar (according to our definition) despite sharing almost no words (they share fewer words in French than in our English translation). On the other hand, CNN-Deep scoring barely better than the Jaccard baseline is consistent with the results reported in [25] for a general-domain corpus. It is striking that, on this data and this task, a tool of this level of sophistication is on par with a very simple overlap measure. The complexity of CNN-Deep's architecture might be ill-suited to the needs of the task at hand. Conversely, the simpler CNN architecture performs better.

Although for each system the differences in accuracy when applied to the different versions of the text are generally small, each system seems to perform best on a specific version. Nonetheless, the three methods do not agree on which level of preprocessing is the most efficient: Jaccard seems to prefer lemmas over stems, while CNN-Deep does the worst on lemmas and the best on stems; and the performance of the CNN deteriorates with any additional processing on top of the initial preprocessing. Surprisingly enough, although stems from unaccented lemmas are more powerful in collapsing the vocabulary, they do not lead to improved performance compared to stems over lemmas

	Question 1	**Question 2**
1	*comment puis je choisir ma place dans le avion* "how can i choose my seat in the plane"	*je souhaiterai savoir comment faire pour réserver un siège en ligne* "i would like to know how to book a seat online"
2	*je ai dépassé la date pour réservé un siège car je pars dans NUMBER jours comment je peus faire* "i have missed the deadline for booking a seat because i leave in NUMBER days what can i do"	*bonsoir puis je réservé mon siege pour cancun* "hello can i book my seat for cancun"
3	*est il possible de payer par chèque* "do you accept cheques"	*peut on payer plusieurs mensualite* "do you accept instalment payments"
4	*quels sont les moyens de paiement* "what are the payment options"	*est il possible de payer par paypal* "can i pay with paypal"
5	*je souhaiterai savoir pourquoi vous ne avez pas de autres dates de disponible pour debut septembre NUMBER* "i would like to know why there are no other dates available for early september NUMBER"	*bonjour je peut pas reserver pour avril* "hello i cannot book for april"
6	*a partir de quand puis je choisir mon siège* "when will i be able to choose my seat"	*j ai reserve et je voudrais savoir ou je suis assise ou si je dois choisir ma place* "i have booked a ticket and i would like to know where i am seated or if i need to choose my seat"
7	*poids* "weight"	*pour un deuxieme bagage vers les dom on a droit a combien de kg* "what is the maximum weight for a second piece of luggage for the overseas departments"
8	*bonjour je suis à la réunion* "i am in réunion"	*cherche vol reunion charles de gaule* "looking for a reunion charles de gaule flight"
9	*je peux prendre le bagage sup sur le vol retour ajout excédents* "can i take the extra baggage on the flight back extra baggage"	*bonjour concernant le bagage supplémentaire quel est le tarif vol cdg neew york jfk classe éco* "hello about the extra baggage what are the fees cdg neew york jfk economy flight"

Table 2. Example QQ pairs from our dataset (preprocessed version). The errors in the French are the users' (*cf.* Section 6.1), and the English translation mimics the French.

as such. Representing the text exclusively as parts of speech has a negative impact on Jaccard and results in an even more marked drop in accuracy for CNN, but does not seem to affect CNN-Deep. Overall, it is hard to assess the benefit of the different types of text preprocessing. Depending on the tool and on the task, the effects may differ.

7.2 Difficulty of the task

The task we set out to tackle is hard. Two human annotators asked to label independently as semantically similar or not a random sample of 100 QQs pairs from our data have achieved a Cohen's kappa as low as 0.332. The annotators were given our definition of semantic similarity and a few examples (including a reply which is so general that it could arguably answer any query), and were instructed to decide whether the two questions in each pair are similar according to the definition. The agreement is very low not only between the annotators, but also between each of them and the ground truth. Very low correlation between raters has been reported in the literature for hard tasks. For instance, on a task that consisted in rating three aspects related to user satisfaction with the dialogue turns of an automated or human interlocutor, [10] reports near-zero Spearman's rank correlation between two raters, including on the easiest of the three aspects, which is deciding if the interlocutor is a good listener or not. Such low agreement may suggest that the task is very hard for a human to solve, that the data may be too noisy for any patterns to be discoverable, or even that there may be no patterns to learn in the data in the first place. We believe the first hypothesis to be plausible in our case. The fact that our best system (CNN) achieves 60% accuracy on the same sample – which, while not as high as the performance on our test sets, is a considerable improvement over random label assignment – points to there being some actual patterns to learn from the data, even if they may not be easily discernible to a human judge.

Specification of the user's information need We believe the difficulty of deciding whether two questions are semantically similar according to our definition may stem from the complexity of correctly inferring the user's real information need from the question they ask. The potential discrepancy between a user's actual information need and what may be inferred from its expression in a textual query is a pervasive problem in information retrieval [21]. As an example, to assess how well suited to a question the answers retrieved by their system were, the authors of [15] had raters "back-generate" a possible information need behind each question before judging the quality of the answers provided by the system. Those researchers point out that for some questions the assessors were unable to reconstruct the original information need, which means they were unable to judge the quality of the answers. Some of the questions in our dataset exhibit an underspecification of the information need (e.g. question 1 in example QQ pair 7), while others are extremely specific (e.g. question 2 in example QQ pair 7); further details are needed about the first one to decide whether the same answer could fit them both. Some questions are incomplete, as question 1 in our example pair 8; if we assume an information need (perhaps the most likely one, or perhaps Paris is the only destination reachable from the origin stated by the user), this question may be viewed as similar to question 2 in pair 8.

Annotators' knowledge of the domain and context Assessing the quality of answers in the domain at hand does not require any technical knowledge, so the "expert/novice" annotator distinction in [19] does not apply here sensu stricto; still, the level of familiarity with the domain (air carrier's products) may affect an annotator's perception of an answer's relevancy. Our annotators were not familiar with the domain, which complicates their assessment of whether the answer is specific enough to satisfy the query. Example QQ pair 2 in Table 2 shows two questions which may very well be acceptably answered by a reply providing comprehensive details about the company's seat reservation policy; however, the first user may expect a reply dealing specifically and exclusively with seat reservation when the deadline has expired. That goes for example pairs 3 and 4: as long as the generic answer is, in fact, exhaustive, it is perfectly valid for any question whose specific answer is included in the generic one. Human raters, however, will find it difficult to decide on the semantic similarity of two questions without some knowledge of the context and the domain. To decide whether the questions in example QQ pair 5 may be similar, one would need to know if both questions were asked a certain number of months earlier than the desired travel date, and if the company does have a policy for handling early bookings, in which case a common answer may satisfy both queries. Likewise, the semantic similarity of example pair 6 may hinge on the actual availability of a seat choice option; if there is none, this will be the valid answer to both questions. QQ pair 9 may be a case of similarity if the company's excess baggage policy is the same regardless of the route.

8 Conclusion and future work

Deciding whether two questions are semantically similar or not is a hard task for humans. Notwithstanding, one of the systems tested in this paper, the CNN, achieved good accuracy on a QQ set derived from user-chatbot exchanges labelled for user satisfaction, outperforming the rule-based chatbot on this task. By simply learning from user-labelled data collected over time, a chatbot can thus improve significantly its ability to detect similar questions in the course of time.

But ultimately, our goal is to assess the usefulness of DQD as part of an answer-retrieving chatbot. Therefore, our next step will be to test our system on Step 2 (*cf.* Section 4), i.e. the actual retrieval of an answer using the output of Step 1 (DQD). To evaluate the performance of our proposed system on this task against the existing system as a baseline, we are preparing a set of questions labelled for their correct answer. Another issue to tackle will be an optimal way of performing fast and efficiently the comparisons between the incoming question and the ones in the reference set as that set grows over time.

In this experimental setup we have restricted ourselves to one-line dialogues, but conversations offer a good ground for yet another application of DQD: detecting the rephrasing of a question during a dialogue, which may be indicative of a problem that requires attention. It would also be interesting to assess the impact of more advanced spelling normalisation and correction on our best system's performance. In addition, new experiments could take account of the known correct answer to a past question when assessing its similarity with a new question. Last but not least, it will be interesting

to validate the results reported here on a similar corpus coming from a different chatbot in a different domain.

Acknowledgements

This research is partly funded by the Regional Council of Brittany through an ARED grant. The present research was also partly supported by the CLARIN and ANI/3279/2016 grants. We are grateful to Telsi for providing the data.

References

1. Accorsi, P., Patel, N., Lopez, C., Panckhurst, R., Roche, M.: Seek&hide: Anonymising a french sms corpus using natural language processing techniques. Lingvisticæ Investigationes 35(2), 163–180 (2012)
2. Afzal, N., Wang, Y., Liu, H.: 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@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016. pp. 674–679 (2016)
3. Baumeister, R.F., Bratslavsky, E., Finkenauer, C., Vohs, K.D.: Bad is stronger than good. Review of general psychology 5(4), 323 (2001)
4. Bernhard, D., Gurevych, I.: Answering learners' questions by retrieving question paraphrases from social Q&A sites. In: Proceedings of the third workshop on innovative use of NLP for building educational applications. pp. 44–52. ACL (2008)
5. Bikel, D.M., Schwartz, R., Weischedel, R.M.: An algorithm that learns what's in a name. Machine learning 34(1), 211–231 (1999)
6. Bogdanova, D., dos Santos, C.N., Barbosa, L., Zadrozny, B.: Detecting semantically equivalent questions in online user forums. In: Proceedings of the 19th Conference on Computational Natural Language Learning, CoNLL 2015, Beijing, China, July 30-31, 2015. pp. 123–131 (2015)
7. Denis, P., Sagot, B.: Coupling an annotated corpus and a lexicon for state-of-the-art pos tagging. Language resources and evaluation 46(4), 721–736 (2012)
8. Feng, M., Xiang, B., Glass, M.R., Wang, L., Zhou, B.: Applying deep learning to answer selection: A study and an open task. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015. pp. 813–820 (2015)
9. Goldberg, Y.: Neural network methods for natural language processing. Morgan&Claypool (2017)
10. Higashinaka, R., Minami, Y., Dohsaka, K., Meguro, T.: Issues in predicting user satisfaction transitions in dialogues: Individual differences, evaluation criteria, and prediction models. In: Spoken Dialogue Systems for Ambient Environments, pp. 48–60. Springer (2010)
11. Hogan, D., Leveling, J., Wang, H., Ferguson, P., Gurrin, C.: Dcu@ fire 2011: Sms-based faq retrieval. In: 3rd Workshop of the Forum for Information Retrieval Evaluation, FIRE. pp. 2–4 (2011)
12. Hone, K.S., Graham, R.: Subjective assessment of speech-system interface usability. In: INTERSPEECH. pp. 2083–2086 (2001)
13. Jalbert, N., Weimer, W.: Automated duplicate detection for bug tracking systems. In: Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on. pp. 52–61. IEEE (2008)

14. Jeon, J., Croft, W.B., Lee, J.H.: Finding semantically similar questions based on their answers. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 617–618. ACM (2005)
15. Jijkoun, V., de Rijke, M.: Retrieving answers from frequently asked questions pages on the web. In: Proceedings of the 14th ACM international conference on Information and knowledge management. pp. 76–83. ACM (2005)
16. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
17. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP. pp. 1746–1751. ACL (2014)
18. Liu, C.W., Lowe, R., Serban, I.V., Noseworthy, M., Charlin, L., Pineau, J.: How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023 (2016)
19. Lowe, R., Serban, I.V., Noseworthy, M., Charlin, L., Pineau, J.: On the evaluation of dialogue systems with next utterance classification. arXiv preprint arXiv:1605.05414 (2016)
20. Malakasiotis, P., Androutsopoulos, I.: Learning textual entailment using svms and string similarity measures. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 42–47. ACL (2007)
21. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
22. Muthmann, K., Petrova, A.: An automatic approach for identifying topical near-duplicate relations between questions from social media q/a sites. In: Proceeding of WSDM 2014 Workshop: Web-Scale Classification: Classifying Big Data from the Web (2014)
23. Reitter, D., Moore, J.D.: Predicting success in dialogue. In: ACL 2007, Proceedings of the 45th Annual Meeting of the ACL, June 23-30, 2007, Prague, Czech Republic (2007)
24. Ritter, A., Cherry, C., Dolan, W.B.: Data-driven response generation in social media. In: Proceedings of the conference on empirical methods in natural language processing. pp. 583–593. ACL (2011)
25. Rodrigues, J.a., Saedi, C., Maraev, V., Silva, J.a., Branco, A.: Ways of asking and replying in duplicate question detection (2017), manuscript accepted for publication
26. Seddah, D., Sagot, B., Candito, M., Mouilleron, V., Combet, V.: The french social media bank: a treebank of noisy user generated content. In: COLING 2012-24th International Conference on Computational Linguistics (2012)
27. Vinyals, O., Le, Q.: A neural conversational model. arXiv preprint arXiv:1506.05869 (2015)
28. Walker, M., Langkilde, I., Wright, J., Gorin, A., Litman, D.: Learning to predict problematic situations in a spoken dialogue system: experiments with how may I help you? In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. pp. 210–217. Association for Computational Linguistics (2000)
29. Walker, M.A., Litman, D.J., Kamm, C.A., Abella, A.: Paradise: A framework for evaluating spoken dialogue agents. In: Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics. pp. 271–280. ACL (1997)
30. Wu, Y., Zhang, Q., Huang, X.: Efficient near-duplicate detection for q&a forum. In: Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011. pp. 1001–1009 (2011)
31. Xue, X., Jeon, J., Croft, W.B.: Retrieval models for question and answer archives. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 475–482. ACM (2008)
32. Yin, W., Schütze, H., Xiang, B., Zhou, B.: Abcnn: Attention-based convolutional neural network for modeling sentence pairs. arXiv preprint arXiv:1512.05193 (2015)