Improving Conversational Question Answering Systems after Deployment using Feedback-Weighted Learning

Jon Ander Campos[1], **Kyunghyun Cho**[2], **Arantxa Otegi**[1],
Aitor Soroa[1], **Gorka Azkune**[1], **Eneko Agirre**[1]
[1]University of the Basque Country (UPV/EHU)
[2]New York University (NYU)
[1] {jonander.campos, arantza.otegi, a.soroa,
gorka.azkune, e.agirre}@ehu.eus,[2]kyunghyun.cho@nyu.edu

Abstract

The interaction of conversational systems with users poses an exciting opportunity for improving them after deployment, but little evidence has been provided of its feasibility. In most applications, users are not able to provide the correct answer to the system, but they are able to provide binary (correct, incorrect) feedback. In this paper we propose feedback-weighted learning based on importance sampling to improve upon an initial supervised system using binary user feedback. We perform simulated experiments on document classification (for development) and Conversational Question Answering datasets like QuAC and DoQA, where binary user feedback is derived from gold annotations. The results show that our method is able to improve over the initial supervised system, getting close to a fully-supervised system that has access to the same labeled examples in in-domain experiments (QuAC), and even matching in out-of-domain experiments (DoQA). Our work opens the prospect to exploit interactions with real users and improve conversational systems after deployment.

1 Introduction

In Conversational Question Answering (CQA) systems, the user makes a set of interrelated questions to the system, which extracts the answers from reference text (Choi et al., 2018). These systems are trained on datasets of human-human dialogues collected using Wizard-of-Oz techniques, where two crowdsourcers are paired at random to emulate the questioner and the answerer. Several projects have shown that it is possible to train effective systems using such datasets. For instance, QuAC includes question and answers about popular people in Wikipedia (Choi et al., 2018), and DoQA includes question-answer conversations on cooking, movies and travel FAQs (Campos et al., 2020). Building such datasets comes at a cost, which limits the widespread use of conversational systems built using supervised learning.

The fact that conversational systems interact naturally with users poses an exciting opportunity to improve them after deployment. Given enough training data, a company can deploy a basic conversational system, enough to be accepted and used by users. Once the system is deployed, the interaction with users and their feedback can be used to improve the system.

In this work we focus on the case where a CQA system trained off-line is deployed and receives explicit binary (correct, incorrect) feedback from users. An example of this task can be seen in Figure 1 where at a point in the conversation two different users give binary feedback to the system according to the correctness of the received answer. Assuming a large number of interactions, we can safely ignore examples for which no feedback is received. We propose feedback-weighted learning (FWL) based on importance sampling as the technique to improve the initial supervised system using only binary feedback from users.

In our experiments user feedback is simulated, and the correct/incorrect feedback is extracted from the gold standard. That is, if the system output matches the gold standard output then it is deemed correct, otherwise it is taken to be incorrect. In order to develop and test feedback-weighted learning we perform initial experiments on document classification. The results show that the model improved by the

Proceedings of the 28th International Conference on Computational Linguistics, pages 2561–2571
Barcelona, Spain (Online), December 8-13, 2020

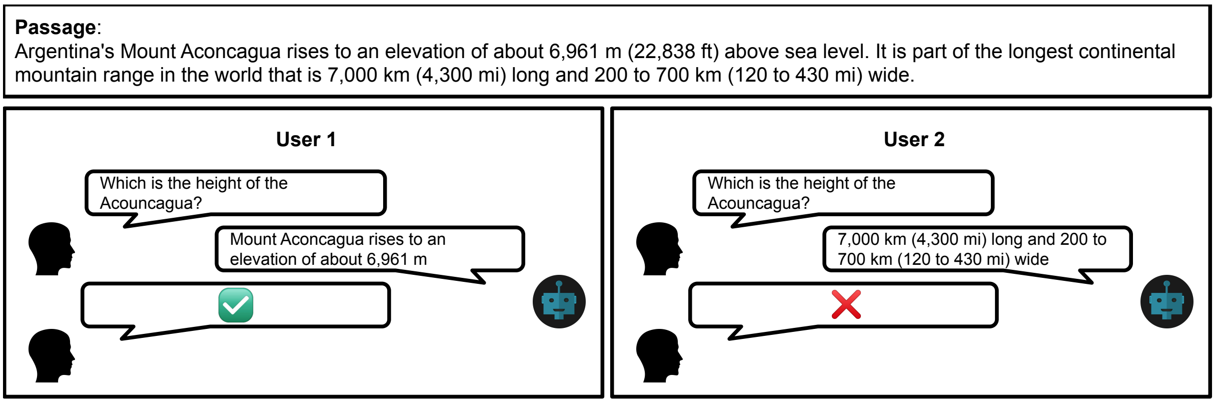

Passage:
Argentina's Mount Aconcagua rises to an elevation of about 6,961 m (22,838 ft) above sea level. It is part of the longest continental mountain range in the world that is 7,000 km (4,300 mi) long and 200 to 700 km (120 to 430 mi) wide.
User 1
Which is the height of the Acouncagua?
Mount Aconcagua rises to an elevation of about 6,961 m
✓
User 2
Which is the height of the Acouncagua?
7,000 km (4,300 mi) long and 200 to 700 km (120 to 430 mi) wide
✗

Figure 1: Example of the CQA task where at a point in the conversation the user 1 gives positive feedback to the system and user 2 gives a negative one due to the received incorrect answer.

proposed algorithm performs comparably to the fully supervised model that is fine-tuned with true labels rather than binary feedback. Those experiments are also used to check the impact of hyperparameters like the weight of the feedback and the balance between exploitation and exploration, which shows that our method is not particularly sensitive to the values of those hyperparameters.

Regarding CQA, we use the best hyperparameters from the earlier experiment on document classification, and conduct experiments using several domains in CQA including datasets like QuAC and DoQA. Our method always improves over the initial supervised system. In the in-domain experiments (QuAC) our method is close to the fully supervised model which is fine-tuned with true labels rather than binary feedback, and in the out-of-domain experiments (DoQA) our method matches it. The out-of-domain results are particularly exciting, as they are related to the case where a CQA system trained off-line in one domain could be deployed in another domain, letting the users improve it via their partial feedback by interacting with the system. Our experiments reveal that the proposed approach is robust to the choice of the system architecture, as we experimented with both multi-layer perceptron and pre-trained transformer.

The main contribution of our work is a novel method based on importance sampling, feedback-weighted learning, which improves the results of two widely used deep learning architectures using partial feedback only. Experimental results from document classification show that feedback-weighted learning improves over the initial supervised system, matching the performance of a fully supervised system which uses true labels. In-domain and out-of-domain CQA experiments show that the proposed method improves over the initial supervised system in all cases, matching a fully supervised system in out-of-domain experiments. This work opens the prospect to exploit interactions with real users and improve conversational systems after deployment. All the code and dataset splits are made publicly available [1].

2 Related Work on Conversational Question Answering

CQA research builds on reading comprehension. In reading comprehension the system has to answer questions about a certain passage of text in order to show that it understands the passage. There are two main methods: the *extractive* method, in which the answer is selected as a contiguous span in the reference passage, and the *abstractive* method, in which the answer text is generated. Many datasets (Rajpurkar et al., 2016; Rajpurkar et al., 2018; Dunn et al., 2017; Kočiskỳ et al., 2018; Trischler et al., 2017; Bajaj et al., 2016) and systems have been proposed to address this task, where the *extractive* scenario has drawn special attention (Wang and Jiang, 2017; Seo et al., 2017). Lately, with the incursion of large pre-trained language models as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and their relatives, the state of the art has been dominated by systems that use the representations obtained with these pre-trained language models. The systems learn answer pointer networks that consist of two

[1] `https://github.com/jjacampos/FeedbackWeightedLearning`

classifiers, one for spotting the start token of the answer span and another for spotting the end token of the answer span. In reading comprehension, the questions are individual and isolated, that is, they do not have any dialogue structure.

Due to the increasing interest on modelling the conversational structure behind user questions, several CQA datasets where questions and answers are interrelated have been created following the Wizard-of-Oz technique. Among all the datasets we can highlight QuAC (Choi et al., 2018), CoQA (Reddy et al., 2019) and DoQA (Campos et al., 2020). While the first two datasets cover more formal domains as Wikipedia articles and literature, the latter covers different domains extracted from online forums as StackExchange. Contextual versions of the previously mentioned reading comprehension models have successfully modelled the conversational structure in those datasets (Qu et al., 2019b; Qu et al., 2019a; Ohsugi et al., 2019; Ju et al., 2019).

3 Importance Sampling for Learning After Deployment

In our learning after deployment scenario we start by training an initial S_0 system in an off-line and supervised way. This first system follows the traditional workflow where we have access to limited supervised training and development data. Then, we take the best performing system on the development data and deploy it to serve user queries. In this deployment phase, every time a user makes a query x, the system generates an answer y and the user gives binary feedback to it. Over time, the system generates different answers $y_{i1}, y_{i2}, ..., y_{in}$ and receives feedback for each item x_i . We assume a sufficient amount of user interactions, and as such we ignore any query-answer pair for which the user did not provide feedback. After the system has been deployed for a while, we collect for each question the answers provided by the system, and the respective user feedback.

We consider a CQA system implemented using two classifiers predicting the start and end tokens respectively. This allows us to consider each classifier independently and describe the process of learning after deployment for a single classifier. We propose to use feedback-weighted learning, which is based on self-normalized importance sampling, in order to generate the system answers.

3.1 Feedback-Weighted Learning

In this section, we describe a novel algorithm for updating a classifier trained off-line on-the-fly based on user feedback alone. We start by defining the true distribution $p^*(y|x)$ over C classes given an input x. This distribution is constructed to reflect binary user feedback $\{-\beta, \beta\}$:

$$p^*(y|x) \propto \begin{cases} \exp(\beta), & \text{if } y \text{ is correct} \\ \exp(-\beta), & \text{if } y \text{ is incorrect} \end{cases}$$

In words, the correctness of each class is reflected in the magnitude of the probability assigned to the class which is proportional to the user feedback. The hyperparameter β controls the weight of the feedback.

The goal of the proposed algorithm is to minimize the KL divergence from p^* to the classifier's predictive distribution $q(y|x;\theta)$ w.r.t. the parameters θ, where

$$\mathrm{KL}(p^*\|q) = -\sum_y p^*(y|x)\log q(y|x;\theta) + \mathcal{H}(p^*). \tag{1}$$

Exact minimization of this objective is however intractable due to the lack of access to the true distribution p^*. We can instead query the unnormalized p^* given the input x and a candidate class y.

We thus resort to self-normalized importance sampling with the following proposal distribution:

$$\hat{q}(y|x) = \lambda q(y|x;\theta) + (1-\lambda)\mathcal{U}(y), \tag{2}$$

where $\mathcal{U}(y)$ is a uniform distribution over y and smooths out the potentially peaky predictive distribution q. We can control this smoothness, which trades off exploration and exploitation, by controlling the mixing coefficient λ (Hoi et al., 2018).

With this proposal distribution, we derive the following objective function for feedback-weighted learning, starting from Eq. (1):

$$\begin{aligned}
\mathrm{KL}(p^* \| q) - \underbrace{\mathcal{H}(p^*)}_{\text{const. w.r.t. } \theta} &= - \sum_y \hat{q}(y|x) \underbrace{\frac{p^*(y|x)}{\hat{q}(y|x)}}_{=w(y^k)} \log q(y|x;\theta) \\
&\approx - \frac{1}{K} \sum_{k=1}^{K} \frac{\omega(y^k)}{\sum\limits_{k=1}^{K} \omega(y^k)} \log q(y^k|x;\theta),
\end{aligned} \tag{3}$$

where K is the total number of user feedback received.

The importance weight $\omega(y^k)$ is computed as

$$\log \omega(y^k) = \underbrace{\beta \mathbb{1}(y^k = y^*)}_{=\text{feedback}} - \log \hat{q}(y|x), \tag{4}$$

where y^* is the (unknown) true class, and

$$\mathbb{1}(\alpha) = \begin{cases} 1, & \text{if } \alpha \text{ is true} \\ -1, & \text{if } \alpha \text{ is false} \end{cases}$$

In other words, the importance weight reflects the ratio between the user feedback and the model's confidence in each sampled prediction y^k. We hence call this algorithm *feedback-weighted learning*.

3.2 Related Work on Lifelong Learning

Continual or lifelong learning is defined as a system's ability to continually learn over time by accommodating new knowledge while keeping previously learned experiences (Parisi et al., 2019). Within this framework of lifelong learning, we particularly focus on building a system that adapts to changes in the data distribution after deployment (Agirre et al., 2019).

There have been efforts for learning actively from dialogue during deployment. The question answering (QA) setting was explored in Weston (2016) and Li et al. (2017), where they analyzed a variety of learning strategies for different dialogue tasks with diverse types of feedback. In these studies they also touch on *forward prediction*, which uses explicit user correction. This idea was later applied to chit-chat systems (Hancock et al., 2019). These works relied on users explicitly providing the correct answer. This strong assumption was relaxed in Weston (2016), where the user provides binary feedback on correct and incorrect answers in a synthetic question answering task (Weston et al., 2015). Our work also uses binary feedback and tests it in more realistic CQA datasets.

In a similar online setup to ours, Liu et al. (2018b) explored contextual multi-armed bandits for dialogue response selection using a customized version of Thompson sampling. In this work they use the Ubuntu Dialogue Corpus (Lowe et al., 2015) for user simulation. In the case of task-oriented dialogue systems, Liu et al. (2018a) propose a hybrid learning method with supervised pre-training and further improvement using human teaching and feedback. For the human teaching case they use imitation learning with explicit corrections done by an expert. After that, they resort to reinforcement learning for further improvement thanks to long term rewards defined by task completion.

4 Experiments

In this section we present the experiments with feedback-weighted learning (FWL). In the experiments we first build a supervised system (S_0), and then we simulate a deployment phase by letting S_0 answer user queries and receiving their feedback. User feedback is derived from a manually annotated deployment set, which is obtained by splitting the training set. We refer to the set used for training S_0 as a *training set* and the other partition of the original training set as a *deployment set* in the rest of the paper.

We consider the following systems and baselines:

- S_0: the original supervised system trained on the training dataset only. We consider this system a baseline.
- S_0 + *FWL*: S_0 is fine-tuned with FWL using examples and partial feedback from the deployment set.
- S_0 + *supervised*: we first train S_0 as above, and then continue its training using examples from the deployment set using the true labels instead of binary feedback. This is thus a fine-tuned system that has full access to the true data.
- *Fully supervised*: a supervised system trained from scratch using the union of the training and deployment sets.

Although our main objective is to develop a lifelong learning system for CQA, we also perform experiments on document classification, as a way to assess the robustness of the proposed method when applied to different neural architectures and tasks. Moreover, these experiments are used to develop the system and check the impact of hyperparameters, so that the best hyperparameters from document classification are used in the CQA experiments.

4.1 Document Classification

The model for document classification is a simple multi layer perceptron (MLP) with a single hidden layer. The input to the MLP is a document vector, calculated as the average of the GloVE vectors (Pennington et al., 2014) of all the words in the document. The dimension of the embeddings is set to 300, and the hidden layer has 200 hidden units.

Experiments are performed on the DBPedia Classes dataset,[2] which contains hierarchical categories of $342,748$ Wikipedia articles. Each article is categorized at three levels into 9, 70 and 219 categories respectively. We use the latter setting with 219 classes in our experiments. The dataset comes with a standard train, development and test splits. We kept the development and test sets untouched, but we split the training part further, creating a training set and a deployment set with the 10% and 90% of the original training examples, respectively. These percentages are motivated on real scenarios where the initial amount of training data is usually limited and expensive to obtain, but during deployment it could be easier to collect more data in a cheaper way. In the deployment phase we consider the feedback to be positive when the class assigned by the system is the same as the gold class in the deployment set, and negative otherwise.

Regarding the experimental setting, the S_0 system is built on the train split using cross entropy loss. For the S_0 + *FWL* system we perform hyperparameter exploration of $\lambda \in [0.5, 1.0]$ and $\beta \in [1, 85]$ using Bayesian optimization (Snoek et al., 2012). The hyperparameter values that performed best in the original development set after one epoch are selected, which corresponds to $\lambda = 0.97$ and $\beta = 76$. We sample class predictions 3 times for each example, based on our preliminary experiments, and train S_0 + *FWL* a maximum of 50 epochs. Given N the amount of training examples and K the amount of samples, in this article we will use *epoch* to mean $N \times K$ feedback requests. See Section 5 for a further discussion on sample efficiency in FWL.

Table 1 shows that the simple MLP architecture performs well on this task, even when only the 10% of training examples are used. Still, S_0 + *FWL* is able to improve the performance of S_0 by 5 points, and it is close to both supervised systems. These results validate the effectiveness of FWL as a way of improving an initial supervised system using binary feedback only.

4.2 Conversational Question Answering

In the CQA experiments we fine-tune a pretrained BERT (Devlin et al., 2019) for QA. Given a query and a passage that contains the answer, the pretrained BERT is fine-tuned to predict the start and end indexes of the answer span. This approach has shown strong performance on QA datasets such as SQuAD

[2] `https://www.kaggle.com/danofer/dbpedia-classes`

Systems	F1
S_0	86.51
S_0 + FWL	91.59 (+5.0)
S_0 + supervised	91.89 (+5.3)
Fully supervised	92.04 (+5.5)

Table 1: Results as F1 on document classification. Number in parenthesis for difference with respect to S_0. FWL continues learning over S_0 using only binary feedback, and the result is close to the supervised systems.

Systems	no history	dialogue history
S_0	46.76	49.03
S_0 + FWL	49.33 (+2.6)	53.07 (+4.0)
S_0 + supervised	53.66 (+6.9)	55.10 (+6.1)
Fully supervised	54.50 (+7.7)	55.40 (+6.5)

Table 2: Results of in-domain experiments using QuAC dataset both for training and deployment, with and without dialogue history. F1 accuracy results on QuAC development split. Number in parenthesis for difference with respect to S_0. FWL is able to improve over S_0 which validates its usefulness in CQA.

(Rajpurkar et al., 2016). In our experiments we use the base uncased model of BERT with the maximum context size of 384 and a batch size of 12, using default values for the rest of the hyperparameters.

We experiment with the following settings:

- In-domain vs. out-of-domain. We experiment with two different scenarios, based on the mismatch between training and deployment distributions. In the first scenario the domain is the same for both training and deployment phases, whereas in the out-of-domain scenario the domains differ.
- Without vs. with dialogue history. In order to take into account the multi-turn feature of a dialogue, we prepend the previous question and its corresponding answer to the input. Following usual practice (Qu et al., 2019a), we consider only the previous interaction (one questions and one answer).

In the in-domain experiments we use QuAC (Choi et al., 2018) for both building the initial S_0 system and during the deployment phase. QuAC is a conversational dataset extracted from the Wikipedia using the Wizard of Oz method and crowdsourcing. In the out-of-domain scenario QuAC is used for building S_0, but the deployment phase is done with DoQA (Campos et al., 2020), which is a conversational dataset based on FAQs and contains dialogues from three different domains (cooking, travel and movies).

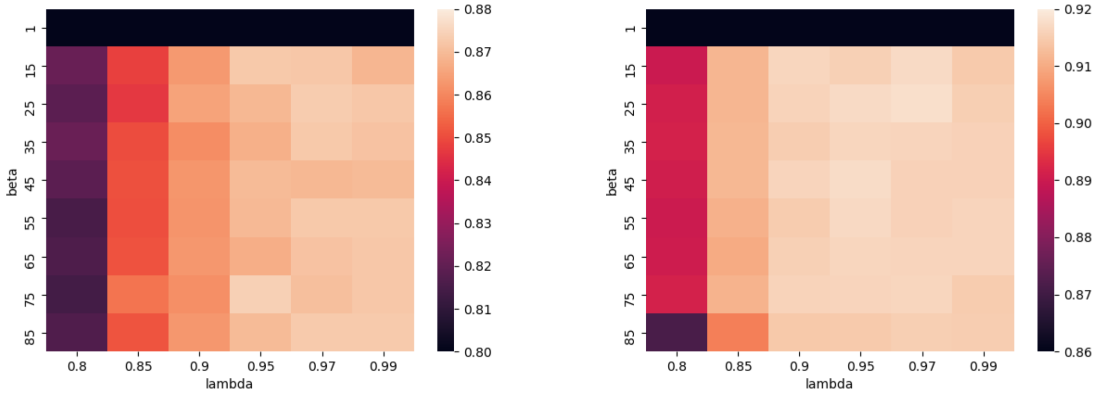
Similarly to document classification, we split the original training parts of QuAC into training and deployment splits containing 10% and 90% of the training dialogues, respectively. We consider the feedback to be positive whenever the answer span predicted by the system matches the gold span exactly, and negative otherwise. Because the QuAC test split is unavailable, we report results in the development split.

With respect to the system settings used for the experiments, we set the λ and β hyperparameters of the S_0 system based on their best values from document classification ($\lambda = 0.97$ and $\beta = 76$). Given that the CQA system contains two classifiers and the number of classes is often larger than in document classification task, we use a larger number of samples, 50 in this task.

Table 2 shows the results on the in-domain experiments on the QuAC dataset. For each system we report the results after 3 epochs following Qu et al. (2019a). The results follow the trend observed in the document classification setting. Applying FWL after S_0 improves the results by 2.6 and 4 points, which confirms that FWL is a valid technique to continue fine-tuning a CQA system after deployment. Using dialogue history improves the results of all systems by almost 3 points, stressing the importance of modeling history on CQA systems. However, the main conclusions remain unchanged. S_0 + *FWL* still outperforms S_0 using only binary feedback, and is close to the supervised systems.

Systems	Cooking	Movies	Travel
S_0	39.79	40.89	35.64
S_0 + FWL	49.66 (+9.9)	47.28 (+6.4)	47.19(+11.6)
S_0 + supervised	50.63 (+10.8)	46.79 (+5.9)	47.12(+11.5)
Fully supervised	50.33 (+10.5)	45.56 (+4.7)	46.10(+10.5)

Table 3: Results of out-of-domain experiments (with history modeling) using QuAC for training and DoQA during deployment. F1 accuracy results on DoQA test split on cooking, movies and travel domains. Number in parenthesis for difference with respect to S_0. FWL improves the results of S_0 and matches supervised results in two domains.



(a) F1 scores obtained after one epoch and using 3 samples

(b) F1 scores obtained after 50 epochs and using 3 samples

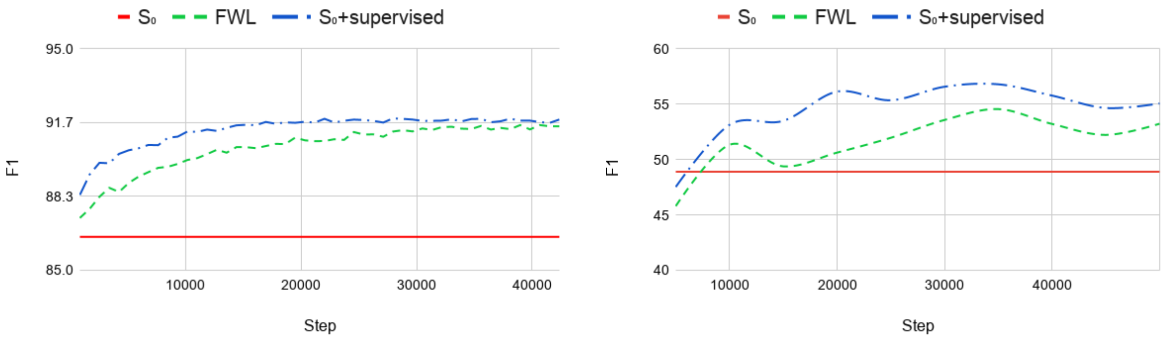
Figure 2: Hyperparameter analysis using heatmaps on document classification showing the obtained F1 scores (lighter is better) in the development split. Similar performance is obtained with different hyperparameter pairs, showing the robustness of the method.

Table 3 shows the results when S_0 is trained on QuAC, and the user feedback is simulated using examples from DoQA. In these experiments we perform model selection on the development split of DoQA (which corresponds to the cooking domain) and report the results on the test datasets comprised of the cooking, travel and movies. We report only experiments using dialogue history, as this setting is more realistic for a CQA system. S_0 + *FWL* outperforms S_0 across all the domains. S_0 + *FWL* furthermore matches the S_0 + *supervised* system in the movies and travel domains, although it fails to do so in the cooking domain. The fully supervised system performs worse than S_0+ *supervised* on this dataset, which we conjecture is due to the fact that QuAC contains more training examples than DoQA, with a ratio of approximately 3 to 1. This may cause the fully supervised system to be more biased towards QuAC, and thus yields worse results in DoQA. Note that in the S_0 + *supervised* system QuAC examples are used to train S_0 only, which is then fine-tuned with DoQA examples, and obtains better results overall. All in all, these results suggest that the FWL approach is robust when there is a domain shift between the training and test datasets.

5 Discussion

As shown by the experiments in document classification and CQA we are able to improve an initial supervised S_0 system just by using binary feedback obtained by simulating the users. In this section we perform a further analysis on several aspacts of the method.

Hyperparameters. In order to show the robustness of FWL we perform several experiments in the document classification task with different values for the main hyperparameters of the method, λ and β (cf. Section 3.1). The analysis shows that when using values larger than 1 for β, FWL performs similarly



(a) Document classification

(b) CQA, in-domain (QuAC dataset), with dialogue history

Figure 3: Learning curves for the document classification and CQA tasks where FWL is compared to supervised learning. As the number of steps increase FWL gets closer to S_0 + *supervised*.

well for all lambdas greater than 0.8 (see Figures 2a and 2b). The behavior of λ in the same Figures 2a and 2b reveals that large values of λ yields best results for all beta values. In any case, the similar performance obtained with different hyperparameter combinations shows that our method is robust and not specially sensitive to small variations in the hyperparameters.

Learning dynamics. From the learning curves in Figures 3a and 3b we see how the behavior is similar in both document classification and CQA learning tasks. In both cases the supervised systems converge faster than the FWL systems, but as the steps go on the F1 scores in the development set also converge. It is of special interest the point where FWL improves over S_0. In the document classification task FWL improves over S_0 in the first steps, and by the end of the first iteration, which comprises circa 850 steps, it already outperforms S_0. In CQA FWL needs more steps but the improvement over S_0 also happens at the beginning of the training process.

Sampling vs. supervised learning. Since we treat epochs in FWL as in supervised learning, we sample new answers for each new epoch. For example, in the document classification case we end up taking 150 samples (50 epochs with 3 samples per epoch) for a total of 219 classes. It can be argued that a dummy sampling technique covering all classes is equivalent to having the true label, and would be similar to our method in terms of sampling efficiency. However, when deploying a $S0$ system in a realistic scenario, the dummy sampling strategy would return low probability responses and could severely hamper user engagement. In contrast, our sampling method tends to return high probability answers, making it more user-friendly. In any case, each time the loss gradient is computed, FWL has information of only 3 samples, unlike supervised learning where all classes are considered. Besides, 3 samples per example (one epoch) are enough for FWL to improve over S_0 (see Figure 2b), although the best results are obtained after 50 epochs.

Assumptions and limitations. We discuss a few assumptions we made in designing the proposed FWL. In all our experiments we simulate user feedback using supervised data, and thus the feedback is always accurate and explicit. We therefore do not consider the case where the user is unsure about the response it gave to the system, which would cause a noisy feedback that can harm the performance of the system. Moreover, as we need more than one sample for each question we would need different users making the same questions if we were to compare our method with real use-cases. Analyzing the impact of these issues and possible solutions to them is kept as an open research question for future analysis.

6 Conclusion and Future Work

In this work we propose feedback-weighted learning that allows a supervised classifier to effectively adapt itself after deployment from partial user feedback. The experiments show that our technique is successful, in that it improves over the initial supervised system. More specifically, in document classification experiments, it matches an off-line supervised system trained with all the true labels, although it has only access to the binary feedback. More importantly, the experiments in two widely used CQA datasets, QuAC and DoQA, confirm that it is feasible to improve a CQA system after deployment. In the DoQA experiments, the CQA system is trained off-line in one domain (Wikipedia) and then deployed in other domains, letting the users improve it via their partial feedback by interacting with the system. In this setting, the performance of our model also matches that of the fully supervised model which is fine-tuned with true labels rather than binary feedback. Moreover, feedback-weighted learning is shown to be effective in two deep learning architectures, including a multi-layer feed forward network and a high-performing pre-trained transformer fine-tuned in the task.

This work uses simulated feedback derived from gold standard labels. In the future we plan to modify feedback-weighted learning to cope with noisy feedback, as well as modifying it to work with fewer samples per query.

Acknowledgments

This research was partially supported by a Google Faculty Award, EU ERA-Net CHIST-ERA LIH-LITH funded by the Agencia Estatal de Investigación (AEI, Spain) project PCIN-2017-118, the Basque Government excellence research group (IT1343-19) and the NVIDIA GPU grant program. Jon Ander Campos enjoys a doctoral grant from the Spanish MECD. Kyunghyun Cho was partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI) and Samsung Electronics (Improving Deep Learning using Latent Structure), and thanks CIFAR, eBay, NVIDIA and NAVER for their support.

References

Eneko Agirre, Anders Jonsson, and Anthony Larcher. 2019. Framing Lifelong Learning as Autonomous Deployment: Tune Once Live Forever. In *International Workshop on Spoken Dialogue Systems Technology*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A Human-Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268*.

Jon Ander Campos, Arantxa Otegi, Aitor Soroa, Jan Deriu, Mark Cieliebak, and Eneko Agirre. 2020. DoQA - Accessing Domain-Specific FAQs via Conversational QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7302–7314, Online, July. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arXiv preprint arXiv:1704.05179*.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from Dialogue after Deployment: Feed Yourself, Chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684.

Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2018. Online learning: A Comprehensive Survey. *arXiv preprint arXiv:1802.02871*.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on Conversational Question Answering. *arXiv preprint arXiv:1909.10772*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA Reading Comprehension Challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Dialogue Learning With Human-in-the-Loop. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. 2018a. Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems. In *Proceedings of NAACL-HLT*, pages 2060–2069.

Bing Liu, Tong Yu, Ian Lane, and Ole J Mengshoel. 2018b. Customized Nonlinear Bandits for Online Response Selection in Neural Conversation Models. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Yasuhito Ohsugi, Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2019. A Simple but Effective Method to Incorporate Multi-turn Context with BERT for Conversational Machine Comprehension. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 11–17.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual Lifelong Learning with Neural Networks: A review. *Neural Networks*, 113:54–71.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019a. BERT with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1133–1136.

Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W Bruce Croft, and Mohit Iyyer. 2019b. Attentive History Selection for Conversational Question Answering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1391–1400.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Shuohang Wang and Jing Jiang. 2017. Machine Comprehension Using Match-LSTM and Answer Pointer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards AI complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv preprint arXiv:1502.05698*.

Jason E Weston. 2016. Dialog-based Language Learning. In *Advances in Neural Information Processing Systems*, pages 829–837.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.