



# Entity Summarization with User Feedback

Qingxia Liu<sup>1</sup>, Yue Chen<sup>1</sup>, Gong Cheng<sup>1(✉)</sup>, Evgeny Kharlamov<sup>2,3</sup>, Junyou Li<sup>1</sup>,  
and Yuzhong Qu<sup>1</sup>

<sup>1</sup> National Key Laboratory for Novel Software Technology, Nanjing University,  
Nanjing, China

{qxliu2013,mg1933005,141220056}@smail.nju.edu.cn,  
{gcheng,yzqu}@nju.edu.cn

<sup>2</sup> Department of Informatics, University of Oslo, Oslo, Norway  
evgeny.kharlamov@ifi.uio.no

<sup>3</sup> Bosch Center for Artificial Intelligence, Robert Bosch GmbH, Renningen, Germany  
evgeny.kharlamov@de.bosch.com

**Abstract.** Semantic Web applications have benefited from entity summarization techniques which compute a compact summary for an entity by selecting a set of key triples from underlying data. A wide variety of entity summarizers have been developed. However, the quality of summaries they generate are still not satisfying, and we lack mechanisms for improving computed summaries. To address this challenge, in this paper we present the first study of entity summarization with user feedback. We consider a cooperative environment where a user reads the current entity summary and provides feedback to help an entity summarizer compute an improved summary. Our approach represents this iterative process as a Markov decision process where the entity summarizer is modeled as a reinforcement learning agent. To exploit user feedback, we represent the interdependence of triples in the current summary and the user feedback by a novel deep neural network which is incorporated into the policy of the agent. Our approach outperforms five baseline methods in extensive experiments with both real users and simulated users.

**Keywords:** Entity summarization · User feedback · Reinforcement learning · Deep neural network

## 1 Introduction

Entity summarization is the task of computing an optimal compact summary for an entity by selecting a size-constrained subset of triples [13]. It has found application in many domains. For example, in Google's Knowledge Graph, an entity may be described by dozens or hundreds of triples. Showing all of them in an entity card would overload users. Google performs entity summarization by selecting key triples that users are likely to need for that particular entity.

An entity summarizer is a tool that computes entity summaries. A wide variety of entity summarizers have been developed [13]. They generate summaries for general purposes [4, 7, 21, 22] or for specific applications such as Web



**Fig. 1.** Two iterations in the cross-replace scenario for entity **Solson Publications**.

browsing [9,23], Web search [10,28], and crowdsourcing [5,6]. However, entity summarization is a difficult task. Recent evaluation results [14] show that summaries generated by existing entity summarizers still differ significantly from ground-truth summaries created by human experts ( $F1 < 0.6$ ). Moreover, current entity summaries are static. There is a lack of mechanisms for improving an entity summary when its quality could not satisfy users’ information needs.

**Research Challenges.** One promising direction to improve entity summarization is to exploit user feedback. This idea has been practiced in related research such as document summarization [1,27] and document retrieval [15,24]. One can establish a cooperative environment where a user reads the current entity summary and conveniently provides feedback to help an entity summarizer compute an improved summary, which in turn is a motivation for user feedback. To effectively incorporate user feedback into entity summarization, there are two research challenges. First, we need to *represent the cooperative process* using a formal model. Second, to exploit user feedback, we need to *represent the interdependence of the current summary and the user feedback*. This is non-trivial because triples have both textual semantics and structural features.

**Contributions.** We address these challenges and study entity summarization with user feedback in the following cross-replace scenario, while our approach can be easily extended to support other scenarios. As illustrated in Fig. 1, a user reads a computed summary  $S_i$  for entity **Solson Publications** and provides negative feedback by crossing off an irrelevant triple  $f_i$ . An entity summarizer analyzes the connection between  $S_i$  and  $f_i$ , and then replaces  $f_i$  with a more relevant triple  $r_i$  to form an improved summary  $S_{i+1}$ . The process can be repeated. To represent this cooperative process, we model an entity summarizer as a reinforcement learning agent, and model the iterative process as a Markov decision process. Further, we represent the interdependence of triples in the current summary and the user feedback by a novel deep neural network which is incorporated into the policy of the agent. Our approach is referred to as DRESSED, short for **Deep Reinforced Entity Summarization with uSer fEeDback**. We carry out a user study to demonstrate the effectiveness of DRESSED. We also conduct extensive offline evaluation based on two benchmarks for evaluating entity summarization and a standard framework of simulating user behavior. DRESSED

outperforms five baseline methods including entity summarizers and relevance feedback models for document summarization/retrieval.

To summarize, our contributions in this paper include

- the first research effort to improve entity summarization with user feedback,
- a representation of entity summarization with iterative user feedback as a Markov decision process,
- a representation of sets of triples and their interdependence as a novel deep neural network, and
- the first empirical study of entity summarization with user feedback based on both real users and simulated users.

The remainder of the paper is organized as follows. We formulate the problem in Sect. 2, and describe our approach in Sect. 3. Online user study and offline evaluation with simulated users are reported in Sects. 4 and 5, respectively. We discuss related work in Sect. 6 before we conclude in Sect. 7.

## 2 Problem Statement

In this section we define the terms used in the paper and formulate the problem.

**Entity Description.** Let  $\mathbf{I}, \mathbf{B}, \mathbf{L}$  be the sets of all IRIs, blank nodes, and literals in RDF, respectively. An RDF dataset  $T$  is a set of RDF triples:

$$T \subseteq (\mathbf{I} \cup \mathbf{B}) \times \mathbf{I} \times (\mathbf{I} \cup \mathbf{B} \cup \mathbf{L}). \quad (1)$$

For triple  $t \in T$ , let  $\text{subj}(t), \text{pred}(t), \text{obj}(t)$  return the subject, predicate, and object of the triple, respectively. The description of an entity  $e$  comprises all the triples in  $T$  where  $e$  is described as the subject or as the object:

$$\text{Desc}(e) = \{t \in T : \text{subj}(t) = e \text{ or } \text{obj}(t) = e\}. \quad (2)$$

**Entity Summarization.** Given an integer size constraint  $k$ , a summary of entity  $e$  is a subset of triples  $S \subseteq \text{Desc}(e)$  such that  $|S| \leq k$ . The problem of entity summarization is to generate an optimal summary from the original entity description by selecting an optimal subset of triples. Optimality could depend on the task and/or the context. We follow most existing researches to generate entity summaries for general purposes.

**User Feedback.** Users and an entity summarizer work in a cooperative environment towards obtaining optimal summaries to best satisfy users' information needs. We consider the following *cross-replace scenario* where a user reads a computed summary and can provide *negative feedback*. Specifically, the summarizer computes and presents a summary  $S_i$  for entity  $e$ . The user reads  $S_i$  and crosses off an irrelevant triple  $f_i \in S_i$ . Based on this negative feedback, the summarizer selects a new candidate triple  $r_i \in (\text{Desc}(e) \setminus S_i)$  to replace  $f_i$  and form an improved summary  $S_{i+1} = (S_i \setminus \{f_i\}) \cup \{r_i\}$ . The process can be repeated until the user provides no further feedback due to satisfaction or loss of patience, or the candidate triples are used up. The problem we study in this paper is how a summarizer *exploits user feedback to identify relevant triples for replacement*.

### 3 Approach

In the cross-replace scenario, an entity summarizer interacts with a user by iteratively exploiting user feedback to compute improved summaries. We want to optimize the user experience during the entire iterative process. It inspires us to model the summarizer as a reinforcement learning agent. Furthermore, an irrelevant triple crossed off by the user should not be presented again. Therefore, the iterative process has states and hence can be modeled as a Markov decision process (MDP), which we will describe in Sect. 3.1. The core problem for an MDP is to find a policy with which the summarizer can exploit irrelevant triples in user feedback to identify relevant triples for replacement. Representing such triple interdependence in a state is non-trivial, for which we propose a novel deep neural network in Sect. 3.2. We solve the learning problem in a standard way and present implementation details in Sect. 3.3. The proposed approach, referred to as DRESSED, is open source under the Apache License.<sup>1</sup>

#### 3.1 Representation of the Cross-replace Scenario

We firstly review MDP and then we model the cross-replace scenario as an MDP.

**MDP.** An MDP is represented as a state-action-reward-transition quadruple denoted by  $\langle \mathcal{Z}, \mathcal{A}, \rho, \tau \rangle$ . An agent interacts with the environment in discrete time steps:  $i = 0, 1, \dots, I$ . At time step  $i$ , the agent is in state  $Z_i \in \mathcal{Z}$ , and follows a  $\theta$ -parameterized policy  $\pi_\theta : \mathcal{A} \times \mathcal{Z} \rightarrow [0, 1]$  to choose an action  $A_i \in \mathcal{A}$  to take. For action  $A \in \mathcal{A}$  and state  $Z \in \mathcal{Z}$ , the policy  $\pi_\theta(A|Z)$  gives the probability of taking action  $A$  when the agent is in state  $Z$ . At time step  $i + 1$ , the agent receives a real-valued immediate reward  $R_{i+1} \in \mathbb{R}$ , and enters state  $Z_{i+1}$ . We assume immediate rewards and state transition can be deterministically characterized by functions  $\rho : \mathcal{Z} \times \mathcal{A} \rightarrow \mathbb{R}$  and  $\tau : \mathcal{Z} \times \mathcal{A} \rightarrow \mathcal{Z}$ , respectively. An iterative process in the cross-replace scenario is represented as a trajectory denoted by  $\xi$ :

$$\xi : Z_0, A_0, R_1 = \rho(Z_0, A_0), Z_1 = \tau(Z_0, A_0), A_1, \dots, R_I, Z_I. \quad (3)$$

The main learning problem here is to find a policy  $\pi_\theta$  that will maximize the expected discounted sum of the immediate rewards over  $\xi$ :

$$J(\theta) = \mathbb{E}_{\xi \sim \pi_\theta} \left[ \sum_{i=1}^I \gamma^{i-1} R_i \right], \quad (4)$$

where  $\gamma \in [0, 1]$  is a discount-rate parameter.

**MDP-Based Modeling.** We model an iterative process in the cross-replace scenario as an MDP. For integer  $i \geq 0$ , let  $S_i$  be the summary computed at time step  $i$ , i.e., in the  $i$ -th iteration. User feedback is part of the environment. Let  $f_i$  be the irrelevant triple crossed off by the user at time step  $i$ . Let  $F_i = \{f_j : 0 \leq j \leq i - 1\}$  represent all the irrelevant triples crossed off prior to time

<sup>1</sup> <https://github.com/nju-websoft/DRESSED>.

step  $i$ . An entity summarizer is an agent. The set of candidate triples for time step  $i$  is  $C_i = \text{Desc}(e) \setminus (F_i \cup S_i)$ . An action  $A_i$  of the summarizer is to select a replacement triple  $r_i \in C_i$  to form an improved summary  $S_{i+1} = (S \setminus \{f_i\}) \cup \{r_i\}$ . We record  $S_i$ ,  $F_i$ ,  $C_i$ , and  $f_i$  in state  $Z_i$ . The full model is defined as follows:

$$\begin{aligned}
 \text{state: } & Z_i = \langle S_i, F_i, C_i, f_i \rangle, \\
 \text{action: } & A_i = r_i, \\
 \text{policy: } & \pi_\theta(t|Z_i) = \frac{\exp(\text{score}(t|Z_i, \theta))}{\sum_{t' \in C_i} \exp(\text{score}(t'|Z_i, \theta))}, \\
 \text{reward: } & R_{i+1} = \rho(Z_i, A_i) = \frac{\text{rel}(r_i)}{\log(i+2)}, \\
 \text{transition: } & Z_{i+1} = \tau(Z_i, A_i) = \langle S_{i+1}, F_{i+1}, C_{i+1}, f_{i+1} \rangle, \\
 \text{initialization: } & Z_0 = \langle S_0, \emptyset, (\text{Desc}(e) \setminus S_0), f_0 \rangle.
 \end{aligned} \tag{5}$$

The policy  $\pi_\theta$  uses a softmax function to map the scores of candidate triples to a probability distribution. Scores are computed by a  $\theta$ -parameterized deep neural network shown in Fig. 2, which we will describe in Sect. 3.2. In the computation of reward during training,  $\text{rel}(r_i)$  is the binary relevance label of triple  $r_i$ : either  $\text{rel}(r_i) = 1$  (relevant) or  $\text{rel}(r_i) = 0$  (irrelevant). We will describe the generation of labeled data in Sect. 3.3.

### 3.2 Representation of Triple Interdependence

The core of our MDP is the representation of policy. A learned policy informs an entity summarizer of how to exploit irrelevant triples in user feedback to identify relevant triples in candidates for replacement. The decision should be conditioned on the current summary and user feedback as well as the user feedback in history. Therefore, the key to the design of our policy in the following is to properly represent all of these triples and their interdependence in a state.

We design a novel deep neural network in Fig. 2 to represent  $\theta$ -parameterized policy  $\pi_\theta$ . All the parameters in the network are collectively referred to as  $\theta$  and will be jointly learned. We rewrite the four elements  $\langle S_i, F_i, C_i, f_i \rangle$  of a state as three sets of triples which are fed into the network as input:

- $S_i \setminus \{f_i\}$ , the set of triples in  $S_i$  that will remain in  $S_{i+1}$ ,
- $F_i \cup \{f_i\}$ , the set of irrelevant triples crossed off till now, and
- $C_i$ , the set of candidate triples for replacement.

Below we detail the four modules of our policy network in Fig. 2. We describe the encoding of a single triple, the encoding of a set of triples, and the encoding of triple interdependence. Based on the encoded triple interdependence in a state as the context, a candidate triple will be selected for replacement.

**Encoding Triples.** For each input triple  $t$ , we jointly encode its textual semantics and structural features using an embedding layer converting  $t$  into a vector representation. Specifically, for each element of  $t$ , i.e., the subject, the predicate,

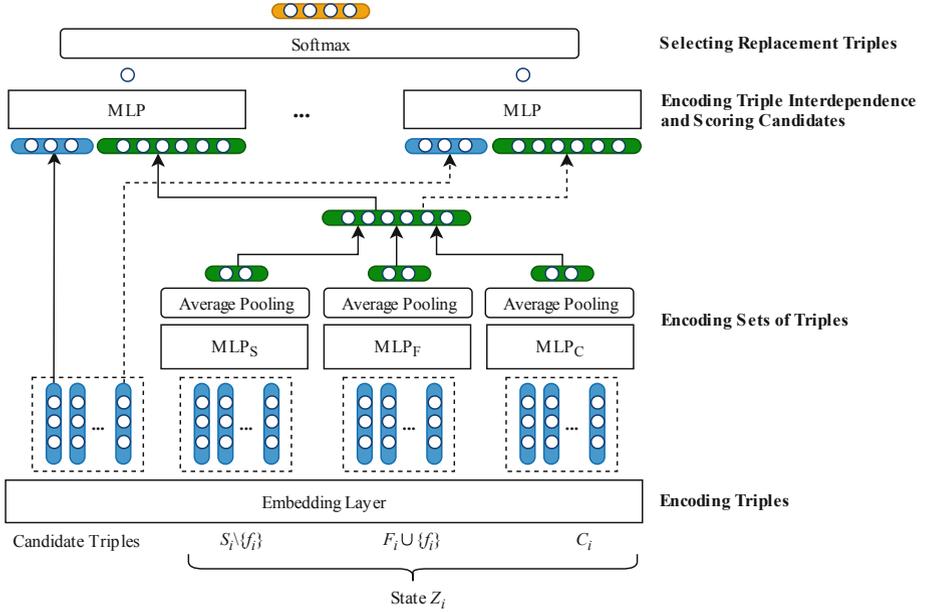


Fig. 2. Policy network.

or the object of  $t$ , we obtain its textual form from its `rdfs:label` if it is an IRI or a blank node, or from its lexical form if it is a literal. We average the pre-trained fastText embeddings [3] for all the words in this textual form as a vector representation of the element to encode its textual semantics. Then we concatenate the vector representations of the three elements of  $t$  to jointly encode its textual semantics and structural features.

**Encoding Sets of Triples.** A state  $Z_i$  is fed into the network as three sets of triples:  $S_i \setminus \{f_i\}$ ,  $F_i \cup \{f_i\}$ , and  $C_i$ . A representation of a set should be permutation invariant to the order of elements. Networks that are sensitive to the order of elements in the input (e.g., RNN) are not suitable. We use a multilayer perceptron (MLP) with two fully connected hidden layers of size 300 and 150, applying Leaky ReLU activations, to process each triple in a set. Then we perform average pooling over all the triples in the set to generate a vector representation for the set, which satisfies permutation invariance. Separate copies of this network (MLP<sub>S</sub>, MLP<sub>F</sub>, MLP<sub>C</sub> in Fig. 2) are used to encode the three input sets. Their vector representations are concatenated to represent state  $Z_i$ .

**Encoding Triple Interdependence and Scoring Candidates.** Finally we encode the interdependence of the three sets in  $Z_i$  and each candidate triple  $t \in C_i$  to score  $t$ . We concatenate the vector representation of  $Z_i$  with the vector representation of  $t$ , and feed the result into an MLP with two fully connected hidden layers of size 64 and 1, applying Leaky ReLU activations. This MLP intercorrelates  $t$  and the three sets of triples in  $Z_i$  to encode their interdependence,

and its output is taken as the score of  $t$ , i.e.,  $\text{score}(t|Z_i, \theta)$  in Eq. (5). The score considers the current summary, user feedback in history, as well as other candidate triples. The scores of all the candidate triples in  $C_i$  are normalized by a softmax layer into a probability distribution, i.e.,  $\pi_\theta(t|Z_i)$  in Eq. (5).

**Selecting Replacement Triples.** One candidate triple in  $C_i$  will be selected as the replacement triple  $r_{i+1}$ . During training, the selection follows the current probability distribution  $\pi_\theta(t|Z_i)$  to address the well-known exploration-exploitation trade-off in reinforcement learning. During testing, exploitation is primary, and hence we greedily select the candidate with the highest probability.

### 3.3 Learning and Implementation

Now we describe our learning algorithm and the generation of labeled data.

**Learning.** To learn an optimal policy  $\pi_\theta$  to maximize  $J(\theta)$  in Eq. (4), we implement REINFORCE [25], a standard policy gradient method in reinforcement learning. Specifically, we update  $\theta$  by computing the following gradient:

$$\nabla_\theta J(\theta) = \gamma^i G_i \nabla_\theta \log \pi_\theta(A_i|Z_i), \quad \text{where } G_i = \sum_{j=i+1}^I \gamma^{j-i-1} R_j. \quad (6)$$

Our implementation uses the Adam optimizer based on TensorFlow with learning rate = 0.01. In Eq. (6), we set the discount rate  $\gamma = 0.6$  to reduce the influence of rewards after 10 iterations below 1% (i.e.,  $0.6^{10-1} \approx 0.01$ ) because users may not be patient with many iterations of interaction.

**Generating Labeled Data.** It is expensive and inflexible to train with real user feedback. We follow a standard synthetic setting in recent information retrieval research [11] to train our model with simulated user behavior. Simulation is based on relevance labels on triples, which can be easily obtained from ground-truth summaries provided by existing benchmarks for evaluating entity summarization such as ESBM [14].

Specifically, for an entity description  $\text{Desc}(e)$  and a ground-truth summary  $S_{\text{gt}}$  thereof, a triple  $t \in \text{Desc}(e)$  is relevant if it appears in  $S_{\text{gt}}$ , otherwise irrelevant. The  $\text{rel}$  function in Eq. (5) is defined accordingly:

$$\text{rel}(t) = \begin{cases} 1 & \text{if } t \in S_{\text{gt}}, \\ 0 & \text{if } t \notin S_{\text{gt}}. \end{cases} \quad (7)$$

We follow a standard framework of simulating user behavior [11] and we adapt it to the cross-replace scenario over the entity summarization task. For entity  $e$ , an initial summary  $S_0$  is generated under size constraint  $k = |S_{\text{gt}}|$  using any standard entity summarizer. Then in the  $i$ -th iteration of the cross-replace scenario, a simulated user: (a) needs to decide whether to provide any feedback, and if so, (b) needs to select an irrelevant triple  $f_i$  from the current summary  $S_i$  to cross off. In our implementation we simulate a perfect user [11] who: (a) will stop providing feedback if and only if  $S_i = S_{\text{gt}}$ , and (b) always

provides noise-free feedback, i.e., never mistakenly crosses off triples in  $S_i \cap S_{\text{gt}}$ . We leave experiments with other user models (e.g., with noise) as future work.

When  $S_i \neq S_{\text{gt}}$ , there may be more than one irrelevant triple in  $S_i$ . Any of them could be crossed off. To let our simulated user behave consistently, we compute and cross off the triple with the highest degree of irrelevance ( $\text{doi}$ ). We learn the  $\text{doi}$  of triple  $t$  by exploiting all the available ground-truth summaries, denoted by  $\mathcal{S}_{\text{GT}}$ . Existing benchmarks such as ESBM usually provide multiple ground-truth summaries created by different human experts for an entity. A triple that appears in fewer ground-truth summaries is more irrelevant. We implement this idea by feeding the vector representation of  $t$  defined in Sect. 3.2 into a two-layer neural network which outputs  $\text{doi}(t) \in [0, 1]$ . We train this network on  $\mathcal{S}_{\text{GT}}$  to minimize the following logistic loss function:

$$- \sum_{S_{\text{gt}} \in \mathcal{S}_{\text{GT}}} \sum_{t \in \text{Desc}(e)} (1 - \text{rel}(t)) \log(\text{doi}(t)) + \text{rel}(t) \log(1 - \text{doi}(t)), \quad (8)$$

where  $S_{\text{gt}}$  is a ground-truth summary for entity  $e$ , and  $\text{rel}$  is defined by Eq. (7).

## 4 Experiment 1: Online User Study

In our first experiment, we carry out a preliminary user study with 24 participants. They are graduate students with at least a basic background in RDF and/or knowledge graphs.

### 4.1 Participating Systems

To the best of our knowledge, DRESSED is the first entity summarizer that can exploit user feedback. We compare it with 2 baselines: a state-of-the-art entity summarizer that cannot exploit user feedback, and a document summarizer that can exploit user feedback and is adapted to perform entity summarization.

**FACES-E** [8] is a state-of-the-art entity summarizer. We obtain its implementation and configuration from its authors. FACES-E relies on UMBC’s Sim-Service which is no longer available. We replace it with a string metric [20]. For entity  $e$ , FACES-E generates a ranking of triples in  $\text{Desc}(e)$  and chooses  $k$  top-ranked triples as a summary. While it cannot exploit user feedback, in each iteration we take the top-ranked candidate triple as the replacement triple.

**IPS** [27] is a popular document summarizer that exploits user feedback to compute an improved document summary by selecting a new set of sentences. To adapt it to entity summarization, we transform each triple into a sentence by concatenating the textual forms of the three elements of the triple. We constrain the search space of IPS such that a re-computed summary differs from the current summary by exactly one triple. This triple will become the replacement triple. Originally, IPS only supports positive feedback. In our implementation we negate the effect of feedback to fit negative feedback in our scenario.

**Training and Tuning.** Following Sect. 3.3, we train DRESSED with simulated user behavior over ground-truth summaries from ESBM v1.0.<sup>2</sup> Each ground-truth summary of an entity consists of 5 triples selected by a human expert from

<sup>2</sup> <https://w3id.org/esbm>.

**Table 1.** Results of online user study (mean  $\pm$  standard deviation). For each method, significant improvements and losses ( $p < 0.01$ ) over other methods are indicated by  $\blacktriangle$  and  $\blacktriangledown$ , respectively. Insignificant differences are indicated by  $\circ$ .

	$I$	$Q_{\text{stop}}$	$Q_{\text{rp1c}}$
FACES-E	$3.26 \pm 2.83$ $\circ\blacktriangle\circ$	$4.25 \pm 0.85$ $\circ\blacktriangle\circ$	$3.51 \pm 1.05$ $\circ\blacktriangledown\blacktriangledown$
IPS	$4.18 \pm 4.99$ $\blacktriangledown\blacktriangledown$	$3.95 \pm 1.05$ $\blacktriangledown\blacktriangledown$	$3.09 \pm 1.22$ $\blacktriangledown\blacktriangledown$
DRESSED	$3.05 \pm 2.70$ $\circ\blacktriangle\blacktriangledown$	$4.25 \pm 0.87$ $\circ\blacktriangle\blacktriangledown$	$3.65 \pm 1.04$ $\blacktriangle\blacktriangle\blacktriangledown$

the original entity description. We set epoch = 100 and batchsize = 16. We also use this dataset to tune the two hyperparameters  $\delta$  and  $\lambda$  of IPS. Their optimal values are found in the range of 0–10 using grid search.

We use CD [26] to generate initial summaries throughout the experiment. However, we could not compare with CD because its output cannot be treated as a ranking of candidate triples like FACES-E.

## 4.2 Procedure and Metrics

For each participant, we randomly sample 35 entities, including 25 entities from DBpedia version 2015-10 and 10 entities from LinkedMDB. Different entities may be assigned to different participants, and they are disjoint from the entities in ESBM which we use for training and parameter tuning. As a within-subject design, for each entity, the participant starts from the initial summary and separately interacts with each summarizer to help to improve the summary. The three summarizers are provided in random order. The experiment is blind, i.e., the participant does not know the order of systems.

In each iteration, the participant is required to cross off an irrelevant triple and then rate the relevance of the replacement triple. This rating,  $Q_{\text{rp1c}}$ , is in the range of 1–5. Participants are instructed to assess relevance with reference to a satisfying general-purpose summary. When the participant decides to stop providing feedback for this entity, s/he rates the quality of the final summary. This rating,  $Q_{\text{stop}}$ , is in the range of 1–5. We also record the number of iterations till termination denoted by  $I$ .

## 4.3 Results

Table 1 presents the results of the online user study. We compare DRESSED with the two baselines and we perform two-tailed t-test to analyze whether their differences are statistical significant ( $p < 0.01$ ).

DRESSED is generally the best-performing approach. First, with FACES-E and DRESSED, participants stop quickly ( $I < 4$ ) and obtain reasonably good summaries ( $Q_{\text{stop}} > 4$ ). The replacement triples selected by the feedback-aware DRESSED during the iterative process are significantly better than those of the feedback-unaware FACES-E according to the results of  $Q_{\text{rp1c}}$ , demonstrating

the usefulness of user feedback and the effectiveness of our approach. Second, compared with DRESSED, participants using IPS perform significantly more iterations but the replacement triples and final summaries they receive are significantly worse. We will justify the performance of these systems in Sect. 5.5.

## 5 Experiment 2: Offline Evaluation

Compared with evaluation with real user feedback, in recent information retrieval research [11] it has been more common to evaluate with simulated user behavior. By conducting this kind of offline evaluation, it would be more achievable and affordable to evaluate many methods at different time steps in varying conditions and, more importantly, the results would be easily reproducible. Our second experiment follows such a standard synthetic setting [11], which has been adapted to the cross-replace scenario over the entity summarization task in Sect. 3.3.

### 5.1 Datasets

As described in Sect. 3.3, simulated user behavior is derived from ground-truth summaries. We obtain ground-truth summaries from the two largest available benchmarks for evaluating entity summarization: ESBM<sup>3</sup> and FED.<sup>4</sup>

ESBM v1.0 provides 600 ground-truth summaries for entities in DBpedia version 2015-10, which we refer to as **ESBM-D**. It also provides 240 ground-truth summaries for entities in LinkedMDB, which we refer to as **ESBM-L**. **FED** provides 366 ground-truth summaries for entities in DBpedia version 3.9. In all these datasets, a ground-truth summary of an entity consists of 5 triples selected by a human expert from the original entity description.

For each dataset, we partition ground-truth summaries and the derived user simulation into 5 equal-sized subsets to support 5-fold cross-validation: 60% for training, 20% for validation, and 20% for test.

### 5.2 Participating Systems

We compare DRESSED with 5 baselines. **FACES-E** [8] and **IPS** [27] have been described in Sect. 4.1. In this experiment we add three relevance feedback models for document retrieval as baselines. For document retrieval, **NRF** [24] is a well-known work that exploits negative relevance feedback, and PDGD [15] represents the state of the art in online learning to rank. Both of them re-rank documents based on user feedback. To adapt them to entity summarization, we transform each triple into a document by concatenating the textual forms of the three elements of the triple. The name of the entity to summarize is treated as a keyword query. After re-ranking, the top-ranked candidate triple is selected as the replacement triple. NRF has three strategies, among which we implement the

<sup>3</sup> <https://w3id.org/esbm>.

<sup>4</sup> <http://wiki.knoesis.org/index.php/FACES>.

SingleQuery strategy. In fact, the three strategies are essentially equivalent in our scenario where user feedback in each iteration is a single triple. For PDGD, we obtain its implementation and configuration from its authors. It has two variants: **PDGD-L** using a linear model and **PDGD-N** using a neural model.

**Training and Tuning.** IPS has two hyperparameters  $\delta$  and  $\gamma$  in the range of 0–10. NRF has three hyperparameters:  $k_1$  in the range of 0–2,  $b$  in the range of 0–1, and  $\gamma$  in the range of 0.5–2. We tune them on the validation set using grid search. PDGD and DRESSED require training. We train their models on the training set. For DRESSED we set epoch = 50 and batchsize = 1.

Initial summaries are generated using CD [26] throughout the experiment.

### 5.3 Metrics

Since we simulate a perfect user, it is meaningless to evaluate the quality of the final summary  $S_I$  which is exactly the ground-truth summary  $S_{\text{gt}}$ . Instead, we evaluate the iterative process, and we use two metrics for different elements of the process: NDCF for summaries, and NDCG for replacement triples.

**NDCF.** Following ESBM, we assess the quality of a computed summary  $S_i$  by comparing it with a ground-truth summary  $S_{\text{gt}}$  and calculating F1:

$$P(S_i) = \frac{|S_i \cap S_{\text{gt}}|}{|S_i|}, \quad R(S_i) = \frac{|S_i \cap S_{\text{gt}}|}{|S_{\text{gt}}|}, \quad F1(S_i) = \frac{2 \cdot P(S_i) \cdot R(S_i)}{P(S_i) + R(S_i)}. \quad (9)$$

Note that in the experiments we have  $P = R = F1$  because  $|S_i| = |S_{\text{gt}}| = 5$ . We evaluate a sequence of summaries  $S_1, S_2, \dots, S_I$  computed during the iterative process. Considering that users will be better satisfied if high-quality summaries are computed earlier, we calculate the normalized discounted cumulative F1 (NDCF) over the first  $i$  iterations ( $1 \leq i \leq I$ ):

$$\text{NDCF}@i = \frac{\sum_{j=1}^i F1(S_j) \cdot \beta^{j-1}}{\sum_{j=1}^i \beta^{j-1}}, \quad (10)$$

where  $\beta \in [0, 1]$  is a discount factor representing the decay of importance. We set  $\beta = 0.6$  to reduce the influence of summaries after 10 iterations below 1% (i.e.,  $0.6^{10-1} \approx 0.01$ ). The result of NDCF is in the range of 0–1. We are particularly interested in  $\text{NDCF}@I$ , which evaluates the entire iterative process.

**NDCG.** We can also assess the quality of a sequence of replacement triples  $r_0, r_1, \dots, r_{I-1}$  selected during the iterative process. We treat the sequence as a (partial) ranking of the triples in  $\text{Desc}(e)$ . Considering that users will be better satisfied if relevant triples are selected earlier, we calculate the normalized discounted cumulative gain (NDCG) of the ranking at position  $i$  ( $1 \leq i \leq I$ ):

$$\text{NDCG}@i = \frac{\text{DCG}@i}{\text{IDCG}@i}, \quad \text{DCG}@i = \sum_{j=1}^i \frac{\text{rel}(r_{j-1})}{\log(j+1)}, \quad \text{IDCG}@i = \sum_{j=1}^i \frac{1}{\log(j+1)}, \quad (11)$$

where  $\text{rel}$  is defined by Eq. (7). NDCG has been widely used in information retrieval. The result is in the range of 0–1. We are particularly interested in  $\text{NDCG}@I$ , which evaluates the entire iterative process.



## 5.4 Results

Table 2 presents the overall results of the offline evaluation. We compare DRESSED with the five baselines and we perform two-tailed t-test to analyze whether their differences are statistical significant ( $p < 0.01$  and  $p < 0.05$ ).

DRESSED significantly ( $p < 0.01$ ) outperforms FACES-E, IPS, and NRF in terms of both  $NDCF@I$  and  $NDCG@I$  on all the three datasets. FACES-E is better than IPS. These results are consistent with the results of our online user study reported in Sect. 4. PDGD-L and PDGD-N are stronger baselines. These latest online learning to rank models achieve better results, but still, DRESSED significantly ( $p < 0.01$ ) outperforms them in terms of both  $NDCF@I$  and  $NDCG@I$  on ESBM-L and FED. However, the difference between DRESSED and PDGD-N in  $NDCF@I$  is not significant ( $p < 0.05$ ) on ESBM-D.

In Fig. 3 and Fig. 4, we plot  $NDCF@i$  and  $NDCG@i$  for  $1 \leq i \leq 10$ , respectively. The results reflect user experience over varying numbers of iterations. DRESSED is consistently above all the baselines in terms of both  $NDCF@i$  and  $NDCG@i$  on all the three datasets. It establishes superiority when  $i$  is very small, i.e., DRESSED better exploits early feedback to quickly improve computed summaries. In particular,  $NDCG@1$  in Fig. 4 indicates the proportion of iterative processes where the first replacement triple is relevant. We observe a very high value  $NDCG@1 = 0.782$  achieved by DRESSED on ESBM-L.

## 5.5 Discussion

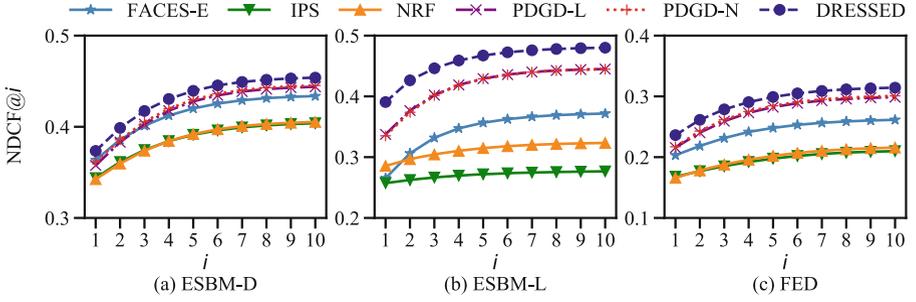
We try to partially justify the performance of the participating systems.

FACES-E is a state-of-the-art entity summarizer but cannot exploit user feedback. Compared with FACES-E, the better performance of DRESSED demonstrates the usefulness of user feedback and the effectiveness of our exploitation.

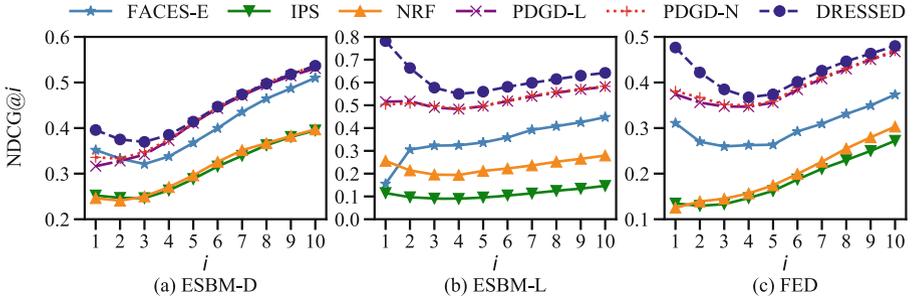
IPS generates a summary having a similar word distribution to the original data. This feature is useful in document summarization but less useful when it is adapted to entity summarization. For example, entity descriptions in LinkedMDB often contain many triples about the **performance** property, which are thus favored by IPS but are rarely included in ground-truth summaries.

NRF relies on word distributions and exact word matching. Such simple text processing techniques are not very effective when processing the textual form of a triple, which can be very short and shows sparsity. By contrast, DRESSED concatenates word embeddings to represent a triple with both textual and structural features to more comprehensively exploit the semantics of the triple.

The two variants of PDGD represent the state of the art in relevance feedback research. Compared with PDGD, one possible reason for the better performance of DRESSED is that we directly and comprehensively model the interdependence of the current summary, user feedback, as well as the user feedback in history, whereas PDGD does not explicitly model all such interdependence.



**Fig. 3.** Results of offline evaluation over varying numbers of iterations (NDCF@ $i$ ).



**Fig. 4.** Results of offline evaluation over varying numbers of iterations (NDCG@ $i$ ).

## 6 Related Work

We discuss three related research topics.

**Entity Summarization.** Entity summarization has been studied for years. RELIN [4] computes informativeness. DIVERSUM [21] improves the diversity of an entity summary by choosing triples with different properties. CD [26] jointly optimizes informativeness and diversity. More than that, FACES [7] and its extension FACES-E [8] consider the frequency of property value, while LinkSUM [22] computes PageRank. ES-LDA [16, 17] relies on a Latent Dirichlet Allocation (LDA) model. However, none of these methods could exploit user feedback to compute improved summaries. From a technical perspective, these methods are unsupervised, whereas our model is based on reinforcement learning. Some application-specific entity summarizers [10, 23, 28] are supervised based on a set of carefully designed features. Our approach avoids such manual feature engineering and learns deep representations of triples and their interdependence.

**Document Summarization with User Feedback.** Some document summarizers exploit user feedback to compute improved summaries. They allow users to select interesting topics [12], keywords [18], or concepts (e.g., named entities) [2]. IPS [27] is the most similar method to our approach. It supports clicking an interesting sentence in a document summary, and leverages this feedback

to compute an improved summary. Compared with unstructured documents, we process structured triples and we encode both the textual semantics and the structural features of a triple. Besides, the above summarizers are unsupervised and stateless, while we model a reinforcement learning agent and we represent the entire iterative process as a Markov decision process.

**Relevance Feedback Models.** Relevance feedback improves the quality of document retrieval based on user-provided relevance judgments. Wang et al. [24] implement a set of methods including the well-known Rocchio algorithm which is based on the vector space model. It modifies the query vector according to user-specified relevant and irrelevant documents. Recent online learning to rank models formulate document retrieval with iterative user feedback as a reinforcement learning problem—usually a dueling bandit problem. A state-of-the-art method is PDGD [15], which constructs a pairwise gradient to infer preferences between document pairs from user clicks. As described in Sect. 5.2, these methods can be adapted to perform entity summarization with user feedback, but their effectiveness may be affected by the adaptation. In document retrieval, there is a query, and the order of the retrieved documents is often used to interpret user feedback. However, in entity summarization, there may not be any query, and the triples in an entity summary can be presented in any order.

## 7 Conclusion and Future Work

We presented the first attempt to improve entity summarization with user feedback. Our reinforcement learning based modeling of the task and, in particular, our deep neural policy network for representing triples and their interdependence, showed better performance than a wide variety of baselines. Our approach has the potential to replace static entity cards deployed in existing applications to facilitate task completion and improve user experience. Our encoder for triples and their interdependence may also find application in other knowledge graph based tasks like entity clustering.

We studied the cross-replace scenario but our implementation can be easily extended to support other scenarios, e.g., crossing off multiple triples in each iteration, crossing without replacement, or providing positive feedback. We will experiment with these extensions in future work. To further improve generalizability, e.g., to deal with paths and structures more complex than triples, one may extend the scope of entity summary with concepts like concise bounded description [19] or RDF sentence [29], to better process blank nodes in RDF.

**Acknowledgments.** This work was supported in part by the NSFC under Grant 61772264 and in part by the Qing Lan Program of Jiangsu Province.

## References

1. Avinesh, P.V.S., Meyer, C.M.: Joint optimization of user-desired content in multi-document summaries by learning from user feedback. In: *ACL 2017*, Volume 1, pp. 1353–1363 (2017). <https://doi.org/10.18653/v1/P17-1124>

2. Avinesh, P.V.S., Meyer, C.M.: Joint optimization of user-desired content in multi-document summaries by learning from user feedback. In: ACL 2017, vol. 1: Long Papers, pp. 1353–1363 (2017). <https://doi.org/10.18653/v1/P17-1124>
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *TACL* **5**, 135–146 (2017)
4. Cheng, G., Tran, T., Qu, Y.: RELIN: relatedness and informativeness-based centrality for entity summarization. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 114–129. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-25073-6\\_8](https://doi.org/10.1007/978-3-642-25073-6_8)
5. Cheng, G., Xu, D., Qu, Y.: C3D+P: a summarization method for interactive entity resolution. *J. Web Sem.* **35**, 203–213 (2015). <https://doi.org/10.1016/j.websem.2015.05.004>
6. Cheng, G., Xu, D., Qu, Y.: Summarizing entity descriptions for effective and efficient human-centered entity linking. In: WWW 2015, pp. 184–194 (2015). <https://doi.org/10.1145/2736277.2741094>
7. Gunaratna, K., Thirunarayan, K., Sheth, A.P.: FACES: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In: AAAI 2015, pp. 116–122 (2015)
8. Gunaratna, K., Thirunarayan, K., Sheth, A., Cheng, G.: Gleaning types for literals in RDF triples with application to entity summarization. In: Sack, H., Blomqvist, E., d’Aquin, M., Ghidini, C., Ponzetto, S.P., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9678, pp. 85–100. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-34129-3\\_6](https://doi.org/10.1007/978-3-319-34129-3_6)
9. Gunaratna, K., Yazdavar, A.H., Thirunarayan, K., Sheth, A.P., Cheng, G.: Relatedness-based multi-entity summarization. In: IJCAI 2017, pp. 1060–1066 (2017). <https://doi.org/10.24963/ijcai.2017/147>
10. Hasibi, F., Balog, K., Bratsberg, S.E.: Dynamic factual summaries for entity cards. In: SIGIR 2017, pp. 773–782 (2017). <https://doi.org/10.1145/3077136.3080810>
11. Jagerman, R., Oosterhuis, H., de Rijke, M.: To model or to intervene: a comparison of counterfactual and online learning to rank from user interactions. In: SIGIR 2019, pp. 15–24 (2019). <https://doi.org/10.1145/3331184.3331269>
12. Leuski, A., Lin, C., Hovy, E.H.: iNeATS: interactive multi-document summarization. In: ACL 2003, pp. 125–128 (2003)
13. Liu, Q., Cheng, G., Gunaratna, K., Qu, Y.: Entity summarization: state of the art and future challenges. CoRR abs/1910.08252 (2019). <http://arxiv.org/abs/1910.08252>
14. Liu, Q., Cheng, G., Gunaratna, K., Qu, Y.: ESBM: an entity summarization benchmark. In: ESWC 2020 (2020)
15. Oosterhuis, H., de Rijke, M.: Differentiable unbiased online learning to rank. In: CIKM 2018, pp. 1293–1302 (2018). <https://doi.org/10.1145/3269206.3271686>
16. Pouriyeh, S.A., Allahyari, M., Kochut, K., Cheng, G., Arabnia, H.R.: Combining word embedding and knowledge-based topic modeling for entity summarization. In: ICSC 2018, pp. 252–255 (2018). <https://doi.org/10.1109/ICSC.2018.00044>
17. Pouriyeh, S.A., Allahyari, M., Kochut, K., Cheng, G., Arabnia, H.R.: ES-LDA: entity summarization using knowledge-based topic modeling. In: IJCNLP 2017, Volume 1, pp. 316–325 (2017)
18. Sakai, H., Masuyama, S.: A multiple-document summarization system with user interaction. In: COLING 2004 (2004)
19. Stickler, P.: CBD - concise bounded description (2005). <http://www.w3.org/Submission/CBD/>

20. Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005). [https://doi.org/10.1007/11574620\\_45](https://doi.org/10.1007/11574620_45)
21. Sydow, M., Pikula, M., Schenkel, R.: The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *J. Intell. Inf. Syst.* **41**(2), 109–149 (2013). <https://doi.org/10.1007/s10844-013-0239-6>
22. Thalhammer, A., Lasierra, N., Rettinger, A.: LinkSUM: using link analysis to summarize entity data. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 244–261. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-38791-8\\_14](https://doi.org/10.1007/978-3-319-38791-8_14)
23. Tonon, A., Catasta, M., Prokofyev, R., Demartini, G., Aberer, K., Cudré-Mauroux, P.: Contextualized ranking of entity types based on knowledge graphs. *J. Web Sem.* **37–38**, 170–183 (2016). <https://doi.org/10.1016/j.websem.2015.12.005>
24. Wang, X., Fang, H., Zhai, C.: A study of methods for negative relevance feedback. In: SIGIR 2008, pp. 219–226 (2008). <https://doi.org/10.1145/1390334.1390374>
25. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992). <https://doi.org/10.1007/BF00992696>
26. Xu, D., Zheng, L., Qu, Y.: CD at ENSEC 2016: generating characteristic and diverse entity summaries. In: SumPre 2016 (2016)
27. Yan, R., Nie, J., Li, X.: Summarize what you are interested. In: An optimization framework for interactive personalized summarization. In: EMNLP 2011, pp. 1342–1351 (2011)
28. Zhang, L., Zhang, Y., Chen, Y.: Summarizing highly structured documents for effective search interaction. In: SIGIR 2012, pp. 145–154 (2012). <https://doi.org/10.1145/2348283.2348306>
29. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on RDF sentence graph. In: WWW 2007, pp. 707–716 (2007). <https://doi.org/10.1145/1242572.1242668>