Task-Aligned Reward Modeling for Reinforcement Learning in Text Summarization: A Survey

Swaroop Nath, Pushpak Bhattacharyya, Harshad Khadilkar
Computation for Indian Language Technology,
Department of Computer Science and Engineering, IIT Bombay, India
{swaroop, harshadk, pb}@cse.iitb.ac.in

Abstract

This survey explores the evolution of learning algorithms for text summarization, focusing on the transition from Supervised Learning (SL) to Reinforcement Learning (RL). Text summarization, the task of generating concise summaries from larger documents, has traditionally relied on SL, which uses labeled datasets to train models by maximizing the likelihood of producing specific outputs given certain inputs. However, SL's limitations in handling the variability and diversity of natural language have prompted a shift towards RL. RL allows for the incorporation of diverse and meaningful rewards, promoting output diversity and semantic similarity. The survey highlights key advancements in RL for text summarization, including the development of several reward functions beyond traditional metrics like ROUGE, such as BERTSCORE for semantic similarity and rewards for factual accuracy. By examining the relevant datasets, evaluation schemes, and emerging trends in reward design, this survey provides a comprehensive overview of the current state and future directions of RL in text summarization. Key challenges and potential areas for further research are also discussed.

1 Introduction

Supervised Learning and Reinforcement Learning have both been used across the spectrum of Natural Language Generation Tasks (Nallapati et al. (2016); Rush et al. (2015); Paulus et al. (2017); Roit et al. (2023), *inter alia*). Supervised Learning (SL) uses labeled datasets, that is, the datasets have both inputs (X_i) and outputs (Y_i); of the form: $\mathcal{D} = \{(X_i, Y_i)\}$. SL trains the model by using the following objective: *maximize the likelihood of observing the output Y_i, given the input X_i*. In essence, SL attempts to train a model to do a task by *showing demonstrations* for the task. On the other hand, Reinforcement Learning (RL) goes beyond this objective, and allows the programmer (developer of the model) to use other rewards. For instance, RL for Text Summarization (the task of automatically generating concise summaries from large documents) has historically used lexical overlap between generated and ground truth summary as a reward (Paulus et al., 2017). Concretely, RL trains the model by using the following objective: *generate such an output Y_i', on the input X_i, which maximizes the obtained reward*. In essence, RL attempts to *encourage* the model to *generate outputs* that maximize the reward. Notice that such an objective removes the constraint of having ground truth outputs (Y_i). We demonstrate the working of SL and RL by taking a *paraphrasing example*[1] below:

An Example. Say we have a dataset with just one instance, with the input (X) and output (Y) given below. Y represents a paraphrased version of X.

X: "*The quick brown fox jumps over the lazy dog.*"

Y: "*The agile brown fox leaps across the sluggish dog.*"

Supervised Learning (SL) trains the *paraphraser model* to maximize the likelihood of Y (*The agile brown fox leaps across the sluggish dog.*) given X, over all possible combinations.

Reinforcement Learning (RL) goes beyond such a training scheme. We can appreciate that there can be several paraphrases for X, all equally valid and correct. RL can help promote this *diversity-based* behaviour, by letting the user choose an appropriate reward function. Say, for this example, we have some reward function (f) which can compute the similarity of meaning between some generation by the paraphraser model Y' and the given input X. RL can use this f to help train the paraphraser model. The paraphraser model generates several

[1]The choice of paraphrasing as an example task is guided by the fact that inputs and outputs are small in size.

paraphrases, which are then rewarded by f. This feedback is used to maximize the likelihoods for generations with high similarity, not just the Y.

Through the example above, we understand a crucial drawback in employing Supervised Learning for Natural Language Generation tasks. There is *inherent variability and diversity in Natural Language*, where the same meaning can be conveyed through different lexical and syntactic constructions. Supervised Learning, with the objective of maximizing the likelihood of a single output for an input, fails to uphold this property of Natural Language. This is a significant reason for exploring Reinforcement Learning in Natural Language Generation tasks.

With the motivation of promoting diversity in outputs, several works have explored Reinforcement Learning (RL) for Natural Language Generation tasks. In the task of Text Summarization, Paulus et al. (2017); Pasunuru and Bansal (2018); Li et al. (2019) are a few representative works. Paulus et al. (2017) explore the usage of ROUGE (which is a lexical overlap based scoring function) as a reward function. Pasunuru and Bansal (2018); Li et al. (2019) provide follow-up works with better reward functions. Both these works stress that ROUGE *is not a sufficiently good reward for Summarization*, as it does not promote properties like semantic similarity, or penalize properties like redundancy within the generated summary. Pasunuru and Bansal (2018) provide an additional reward Saliency that checks if the summary incorporates important aspects. Li et al. (2019) propose the usage of BERTSCORE as an additional reward, to promote semantic similarity. In a similar way, Roit et al. (2023); Tang et al. (2023) attempted to reward factuality within generated summaries.

In this survey we look at how Learning algorithms have evolved for Text Summarization, from Supervised Learning to Reinforcement Learning. This has promoted a shift in focus from searching for better and bigger architectures, to designing better rewards for the Reinforcement Learning algorithm. The survey is structured as follows: Section 2 introduces the Text Summarization tasks, Section 4 provides a listing of the relevant datasets, Section 5 highlights the popular evaluation schemes used in these tasks, Section 6 introduces some Supervised Learning methods, Section 7 highlights the recent emergence of Reinforcement Learning methods, and discusses the trends in the Rewards used, Section 8 highlights the challenges existent in current approachs, and finally Section 9 summarizes our Survey.

2 Background: Text Summarization

Text Summarization is a field of research that aims to generate a concise and coherent summary of a single document or multiple documents. The goal is to produce a summary that captures the most important information from the source text, providing a clear and brief overview without requiring the reader to go through the entire document.

A practical example includes:

Document[2] - *Climate change affects the social and environmental determinants of health – clean air, safe drinking water, sufficient food and secure shelter. Between 2030 and 2050, climate change is expected to cause approximately 250,000 additional deaths per year, from malnutrition, malaria, diarrhea and heat stress. ⋯ The direct damage costs to health are estimated to be between USD 2-4 billion per year by 2030. ⋯ Areas with weak health infrastructure – mostly in developing countries – will be the least able to cope without assistance to prepare and respond. Reducing emissions of greenhouse gases through better transport, food and energy-use choices can result in improved health, particularly through reduced air pollution.*

Summary - *Climate change impacts health by affecting air, water, food, and shelter. It is expected to cause 250,000 additional deaths annually between 2030 and 2050 and cost USD 2-4 billion per year in health damages. Developing countries with weak health infrastructure are most vulnerable. Reducing greenhouse gas emissions can improve health by decreasing air pollution.*

Text summarization has significant applications and potential impacts across various domains. In news aggregation, it helps users quickly grasp key points of articles, making information consumption more efficient. In document management, summarization assists in handling large volumes of text, enabling quicker access to essential information. In scientific research, it allows researchers to stay updated with the latest findings by providing concise summaries of academic papers.

According to Statista (Figure 1), the global data volume is projected to grow from 33 zettabytes in 2018 to 181 zettabytes by 2025. This exponential

[2]The document has been taken from who.int/news-room/climate-change-and-health

increase highlights the necessity for efficient information processing tools like text summarization. Summarization can manage this information overload, enhancing productivity and decision-making. By distilling essential information from vast texts, summarization tools make information more accessible and actionable, significantly impacting education, healthcare, business, and beyond.

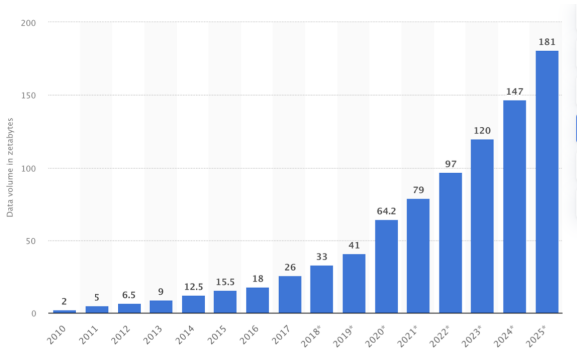

Figure 1: Growth of the volume of digital data (in zettabytes) from 2010 to 2025 (forecasted). We can see that the volume crossed the 100 zettabytes mark in 2023 (1 zettabyte = 1 billion terabyte). Source: statista.com/digital-data

In this survey work, we look at two specific kinds of summarization — (*a*) **Query-focused Summarization** (Section 2.1), and (*b*) **Opinion Summarization** (Section 2.2).

2.1 Query-focused Summarization

Query-focused Summarization (QfS) is a specialized field of research withing Text Summarization that aims to generate a summary from a single document or multiple documents based on a provided query. This type of summarization is particularly useful when dealing with open-ended questions, which require more comprehensive and context-specific responses compared to straightforward factual queries. The input and output are concretely stated below:

Input: *Query* and *Document*.

Output: A *summary* pertaining to the query, from the document.

A practical example of QfS is as follows:

Query: *What is String Pool in Java?*

Document[3]: *String pool is nothing but a storage area in Java heap where string literals store. It is also known as String Intern Pool or String Constant Pool. ⋯ It is just like object allocation. By default, it is empty and privately maintained by the Java String class. Whenever we create a string, the string object occupies some space in the heap memory. Creating a number of strings may increase the cost and memory too, which may reduce the performance. ⋯ The JVM performs some steps during the initialization of string literals that increase the performance and decrease the memory load. To decrease the number of String objects created in the JVM, the String class keeps a pool of strings. When we create a string literal, the JVM first checks that literal in the String pool. ⋯ If the literal is already present in the pool, it returns a reference to the pooled instance. If the literal is not present in the pool, a new String object takes place in the String pool.*

Summary: *A String Pool in Java is a specific area in memory allocated to store string literals. It optimizes memory usage and improves performance by reusing existing string instances.*

Query-focused Summarization has several important applications, making it highly valuable in various domains:

- **Customized Search Engine Results**: Such systems can be deployed to produce coherent summaries of top-ranked websites based on the user's query. Companies like Google and You.com are exploring this application to enhance search engine results by providing more relevant and concise information.
- **Summarization in Conversational Question Answering (CoQA)**: Exploring summarization capabilities in Conversational Agents has been a long-standing research interest. Recent surge of Large Language Model based chatbots (ChatGPT, OpenAI (2023); Gemini, DeepMind (2023); Claude, Anthropic (2023)) has shown the importance of Query-focused Summarization within Conversational Agents.

2.2 Opinion Summarization

Opinion Summarization is another specialized field of research within Text Summarization that aims to generate a coherent summary of opinions expressed towards a product (such as phones, laptops, books, etc.) or a service (such as movies). This type of summarization is particularly useful for aggregating and condensing subjective viewpoints, such as product reviews, customer feedback, or social media posts. The input and output are concretely stated below:

[3]The document has been taken from javatpoint.com/string-pool-in-java

Input: *Reviews* containing opinions about a product/service from the users.
Output: A *summary* that encapsulates the overall sentiment and key points from the reviews.

A practical example of Opinion Summarization is as follows:

Input Reviews:

> **Review 1**: *The new smartphone model has a fantastic battery life and an impressive camera. However, the screen size is too large for my liking.*
>
> **Review 2**: *I love the sleek design and the battery life is outstanding. But, the phone is a bit too expensive for the features it offers.*
>
> **Review 3**: *Great camera quality and the battery lasts all day. The large screen makes it difficult to use with one hand though.*

Opinion Summary: *The new smartphone is praised for its excellent battery life and camera quality. However, users find the large screen size inconvenient and consider the phone to be overpriced.*

Opinion Summarization has several important applications, making it highly valuable in various domains:

- **Product Reviews Analysis**: This system can be deployed to aggregate and summarize customer reviews on e-commerce platforms. For example, it can provide a comprehensive summary of user opinions on a new product, helping potential buyers make informed decisions.
- **Customer Feedback Insights**: Companies can use opinion summarization to analyze customer feedback from surveys or support tickets, identifying common themes and areas for improvement. This can significantly enhance customer satisfaction by addressing frequent issues and improving services.
- **Social Media Monitoring**: In the realm of social media, opinion summarization can help track public sentiment about brands, products, or events. It can condense vast amounts of social media posts into a digestible format, allowing businesses to quickly understand public perception and react accordingly.

By distilling subjective content into concise and insightful summaries, Opinion Summarization enhances the accessibility and utility of opinion data, providing actionable insights that can drive better decision-making across various sectors.

3 Background: Reinforcement Learning

Reinforcement Learning (RL) is a framework for learning optimal decision-making strategies through interaction with an environment. In RL, an agent takes actions in an environment and receives feedback in the form of rewards or penalties, which it uses to learn a policy that maximizes cumulative rewards over time. We describe the necessary terminologies below. This section formalizes the key concepts of Reinforcement Learning, including Markov Decision Processes (MDPs), Value Functions, and Policy Optimization.

Necessary Terminologies. Before diving into the depths of Reinforcement Learning, let us look at a few important definitions that are frequently observed in the literature:

***Agent*:** This is the entity we are trying to train to take rational and intelligent decisions. For example, in the context of text summarization, the Summarizer Model is the *Agent*.

***Environment*:** Environment is the surrounding that the agent interacts with. Environment is what provides a reward/penalty to the agent. For example, in the context of text summarization, the Input Document and the Oracle that judges the summary form the *Environment*.

***State*:** State defines the current situation the agent is in. For example, in the context of text summarization, the Partially Generated Summary at any time-step t and the Input Document form the *State*.

***Action*:** Action is the choice that the agent takes while in a state. This takes the agent from one state to another. For example, in the context of text summarization, the Generated Token represents the *Action*.

***Policy*:** Policy is the thought process that the agent follows to take an action while in a state.

***Trajectory*:** Trajectory is the sequence of actions taken by the agent. For example, in the context of text summarization, the Generated Summary represents the *Trajectory*.

3.1 Markov Decision Processes

A Markov decision process (MDP) is a formal framework for modeling sequential decision-making problems. An MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ is the state space, representing all possible states of the environment.
- $\mathcal{A}$ is the action space, representing all possible actions the agent can take.
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, which specifies the probability of transitioning from one state to another given an action.
- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, which maps state-action pairs to immediate rewards.
- $\gamma \in [0, 1]$ is the discount factor, which determines the importance of future rewards relative to immediate rewards.

The goal in an MDP is to find a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maps states to actions, maximizing the expected cumulative reward over time.

3.2 Value Functions

Value functions provide a way to evaluate the quality of states and state-action pairs under a given policy. The state-value function $V^\pi(s)$ represents the expected cumulative reward starting from state s and following policy π:

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, \pi\right]$$

where $a_t \sim \pi(s_t)$ and $s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$. The action-value function $Q^\pi(s, a)$ represents the expected cumulative reward starting from state s, taking action a, and following policy π:

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a, \pi\right]$$

Value functions satisfy recursive relationships known as Bellman equations. For the state-value function, the Bellman equation is:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \Big(R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V^\pi(s') \Big)$$

For the action-value function, the Bellman equation is:

$$Q^\pi(s, a) = R(s, a) + \gamma \Big(\sum_{s' \in \mathcal{S}} P(s'|s, a) \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^\pi(s', a') \Big)$$

3.3 Policy Optimization

Policy Optimization involves finding the policy that maximizes the expected cumulative reward. One common approach is policy iteration, which alternates between policy evaluation and policy improvement. In policy evaluation, the value function under the current policy is computed using the Bellman equations. In policy improvement, the policy is updated to be greedy with respect to the current value function:

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} Q^\pi(s, a)$$

This process iterates until convergence, resulting in an optimal policy π^* that maximizes cumulative rewards.

Another approach is value iteration, which directly updates the value function using the Bellman optimality equation:

$$V^*(s) = \max_{a \in \mathcal{A}} \left(R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V^*(s') \right)$$

The optimal policy is then derived from the optimal value function:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \left(R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V^*(s') \right)$$

3.4 Exploration vs. Exploitation

A fundamental challenge in Reinforcement Learning is the exploration-exploitation trade-off, where the agent must balance between exploring new actions to discover potentially better policies and exploiting known actions to maximize immediate rewards. Strategies such as epsilon-greedy exploration and upper confidence bound (UCB) methods are commonly used to address this trade-off (Sutton and Barto, 2018).

In epsilon-greedy exploration, the agent selects a random action with probability ϵ and the best-known action with probability $1 - \epsilon$. This ensures a balance between exploration and exploitation:

$$a_t = \begin{cases} \text{random action} & \text{with probability } \epsilon \\ \arg\max_{a \in \mathcal{A}} Q(s_t, a) & \text{with probability } 1 - \epsilon \end{cases}$$

Upper confidence bound (UCB) methods select actions based on both their estimated value and the uncertainty in their estimates, encouraging exploration of less certain actions:

$$a_t = \arg\max_{a \in \mathcal{A}} \left(Q(s_t, a) + c\sqrt{\frac{\log t}{N(s_t, a)}} \right)$$

where c is a constant controlling the degree of exploration, t is the time step, and $N(s_t, a)$ is the number of times action a has been taken in state s_t.

3.5 Examples of Reinforcement Learning Applications

Reinforcement Learning has a wide range of applications, demonstrating its versatility and effectiveness in solving complex decision-making problems:

1. *Game Playing*: RL has been successfully applied to games, such as AlphaGo (Silver et al., 2016), where the agent learns to play the game at a superhuman level.

2. *Robotics*: RL is used in robotics for tasks like robotic arm manipulation and autonomous driving, where the robot learns to perform tasks through trial and error.

3. *Summarization*: RL has beeen used to improve text summarization (Paulus et al., 2017) by optimizing the summary based on reward signals derived from human feedback or specific summarization metrics.

Reinforcement Learning offers a powerful paradigm for developing intelligent agents capable of learning from interaction with their environment. By leveraging mathematical formulations and algorithms, RL enables agents to make sequential decisions that maximize long-term rewards.

Reward Hypothesis (Sutton and Barto, 2018; Silver et al., 2021) is a foundational principle in Reinforcement Learning, positing that *all goals and tasks can be encapsulated by a reward function*. Formally, this hypothesis asserts that the objective of an agent is to maximize the expected cumulative reward, which is defined by a scalar signal received from the environment. The reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ assigns a numerical value to each state-action pair, guiding the agent towards desirable behaviors. In our work, we categorize the rewards into three categories: (a) those that are computable and differentiable, (b) those that are computable but non-differentiable, and (c) those that are non-computable. This categorization is helpful for Natural Language Generation tasks.

3.5.1 Computable and Differentiable Rewards

Computable and Differentiable reward functions can be explicitly calculated and have smooth gradients, making them amenable to gradient-based optimization methods.

- **Robotics Example**: Consider a robot tasked with reaching a target location. The reward function could be defined as the negative Euclidean distance to the target:

  $$R(s, a) = -\|s_{\text{current}} - s_{\text{target}}\|$$

  where s_{current} is the robot's current position and s_{target} is the target position. This reward function is both computable and differentiable, allowing for the application of methods like gradient descent to optimize the robot's trajectory.

- **Summarization Example**: In text summarization, a differentiable reward might be based on the cosine similarity between the generated summary and a reference summary, embedding both using a pre-trained language model:

  $$R(s, a) = \cos(\text{emb}(s_{\text{generated}}), \text{emb}(s_{\text{reference}}))$$

  Here, $\text{emb}(\cdot)$ represents the embedding function, and $s_{\text{generated}}$ and $s_{\text{reference}}$ are the embeddings of the generated and reference summaries, respectively (Zhang et al., 2019).

3.5.2 Computable and Non-Differentiable

Computable but Non-Differentiable rewards are explicitly calculable but lack smooth gradients, posing challenges for gradient-based methods.

- **Game Playing Example**: In a board game, the reward might be $+1$ for a win, 0 for a draw, and -1 for a loss:

  $$R(s, a) = \begin{cases} 1 & \text{if win} \\ 0 & \text{if draw} \\ -1 & \text{if loss} \end{cases}$$

  This reward is computable at the end of the game but non-differentiable, as it provides discrete feedback.

- **Summarization Example**: Using ROUGE-L, which measures the longest common subsequence between the generated summary and the reference, provides a computable but non-differentiable reward:

$$R(s, a) = \text{ROUGE-L}(s_{\text{generated}}, s_{\text{reference}})$$

This metric is computable but requires alternative optimization techniques such as REINFORCE to handle non-differentiability (Paulus et al., 2017).

3.5.3 Non-Computable

Non-computable rewards are those that cannot be directly calculated from state-action pairs and often involve subjective or complex evaluations.

- **User Satisfaction Example**: In recommendation systems, the reward might be based on user satisfaction, which is inherently subjective and cannot be directly computed from the system's states and actions. Surveys or feedback mechanisms are often used to approximate this reward.
- **Summarization Example**: Human evaluations of summary quality, coherence, and informativeness fall into this category. These evaluations are often gathered using Likert scales or comparative assessments (e.g., Best-Worst Scaling):

$$R(s, a) = \text{LikertScore}(s_{\text{generated}})$$

or

$$R(s, a) = \text{ComparativeScore}(s_{\text{generated}}, s_{\text{reference}})$$

These methods rely on human judgement, which can be subjective and variable (Bhandari et al., 2020).

3.6 Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) is an approach where an agent learns optimal behavior through evaluative feedback provided by humans. This method is particularly useful for tasks where it is difficult to specify a reward function explicitly (non-computable rewards). Instead of relying on a predefined reward signal, RLHF leverages human judgments to dynamically shape the reward function.

Mathematical Formulation. In the RLHF framework, the standard reinforcement learning setup is modified to include human feedback. Consider a partial Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma)$, where:

- $\mathcal{S}$ is the set of states,
- $\mathcal{A}$ is the set of actions,
- P is the state transition probability function $P(s'|s, a)$,
- γ is the discount factor.

In RLHF, the reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is learned from human feedback rather than being predefined. The human feedback is used to train a reward model $\hat{R}$ that approximates the true underlying reward function. Let $\mathcal{D} = \{(s_i, a_i, r_i)\}_{i=1}^{N}$ be a dataset of state-action pairs and their corresponding rewards provided by humans. The objective is to minimize the error between the predicted rewards and the human-provided rewards:

$$\min_{\hat{R}} \mathbb{E}_{(s,a,r)\sim\mathcal{D}} \left[\left(\hat{R}(s, a) - r \right)^2 \right]$$

Once the reward model $\hat{R}$ is learned, we have the full MDP: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \hat{R}, \gamma)$. It is now used within the Reinforcement Learning framework to optimize the policy π. The policy $\pi : \mathcal{S} \to \mathcal{A}$ aims to maximize the expected cumulative reward as given by the learned reward model:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \hat{R}(s_t, a_t) \right]$$

Examples. We provide a few examples below, where RLHF can benefit the policy learning process:

1. *Robotics*: In robotic manipulation tasks, defining a precise reward function can be challenging. Human feedback, in the form of binary success/failure signals or scalar ratings on task performance, can be used to train a reward model. For example, a human might provide feedback on how well a robot grasps and moves objects, enabling the robot to refine its actions based on these evaluations.
2. *Summarization*: In the context of text summarization, human feedback can be used to train

models to generate more coherent and relevant summaries (Ziegler et al., 2019). Humans can provide feedback on generated summaries by rating them on aspects such as informativeness, fluency, and relevance. This feedback can be used to train a reward model $\hat{R}$, which the summarization model uses to optimize its policy for generating better summaries.

Learning from Comparisons. An effective approach within RLHF is to use human comparisons to derive rewards. In this setup, humans are presented with two or more options and asked to indicate which one is better. These comparisons can be used to infer a reward model by training on pairwise preferences.

Formally, given pairs (s_i, a_i) and (s_j, a_j) with feedback indicating a preference for (s_i, a_i) over (s_j, a_j), we can use a logistic regression model to learn the reward function:

$$P((s_i, a_i) \succ (s_j, a_j)) = \frac{\exp(\hat{R}(s_i, a_i))}{\exp(\hat{R}(s_i, a_i)) + \exp(\hat{R}(s_j, a_j))}$$

The objective is then to maximize the likelihood of observed human preferences:

$$\max_{\hat{R}} \sum_k \log P((s_{i_k}, a_{i_k}) \succ (s_{j_k}, a_{j_k}))$$

By incorporating human feedback in this manner, RLHF allows for the development of more flexible and adaptive agents capable of learning complex behaviors that are aligned with human expectations and preferences.

4 Datasets

4.1 Query-focused Summarization

The availability and size of datasets have significantly influenced research in QfS. Initial datasets like DUC 2005 and DUC 2006 (Dang, 2005, 2006) provided a foundational benchmark but were limited in size, making them insufficient for training large neural models.

To address this limitation, Nema et al. (2017) introduced the DebatePedia dataset, which consists of 12,695 samples, providing a larger scale resource for training neural models. Despite this advancement, the need for even larger datasets remains.

Several notable datasets for QfS include:

DebatePedia: Contains 12,695 samples designed for QfS tasks, facilitating the development of more robust neural models (Nema et al., 2017).

Natural Questions: A dataset by Kwiatkowski et al. (2019) initially created for extractive question answering but applicable for extractive QfS due to its inclusion of long-form answers.

ELI5: Presented by Fan et al. (2019), this dataset, though designed for long-form question answering (LfQA), is suitable for QfS as it provides detailed responses to specific queries. The ELI5 dataset, particularly the version not included in the KILT framework (Petroni et al., 2021), is utilized due to its provision of gold documents for summary generation.

QMDSCNN and **QMDSIR**: Introduced by Pasunuru et al. (2021), these datasets are designed for multi-document abstractive QfS, providing new avenues for research in this area.

RQFT: Reliable QfS Tester (RQFT) was introduced by Nath et al. (2023) as a benchmark for testing QfS solutions. The proposed dataset contains human queries on topics from high-school text books and Wikipedia articles, containing 250 instance. The dataset also tackles topic centralization (Baumel et al., 2016), a common drawback in several QfS benchmarks.

4.2 Opinion Summarization

Flipkart (Siledar et al., 2023b): Flipkart dataset contains product reviews from three domains: *laptops, mobiles*, and *tablets*. The test set contains around 147 products with one summary per product. Each summary consists of multiple aspect-specific summaries. There are around 676 aspect-specific summaries in total. The original test set contains around 1000 reviews per product on average. Siledar et al. (2023a) downsample this to 10 reviews per product to compare different models. They first remove all the reviews with less than 20 and more than 100 words. For filtering out 10 reviews they use a simple approach of first checking if the reviews contain the aspects for which summaries need to be created. After the filtering step, they randomly selected 10 reviews to form input for our test set.

GPT-R/GPT-RDQ (Siledar et al., 2024) extended the already available Amazon, Oposum+, and Flipkart test sets by leveraging ChatGPT for annotation. GPT-R used only reviews while generat-

ing the summary whereas GPT-RDQ used reviews, description, and question-answers for generating summaries. They curated 6 new test sets: Amazon GPT-R, Amazon GPT-RDQ, Oposum+ GPT-R, Oposum+ GPT-RDQ, Flipkart GPT-R, and Flipkart GPT-RDQ containing 662 opinion summaries in total.

AmaSum (Bražinskas et al., 2021): The AmaSum dataset is a large-scale abstractive opinion summarization dataset containing over $33,000$ human-written summaries for Amazon products. Each summary is paired with more than 320 customer reviews and includes three types of summaries: verdict, pros, and cons.

Space (Angelidis et al., 2021): The Space dataset is a large-scale benchmark for evaluating unsupervised opinion summarizers, built on TripAdvisor hotel reviews. It includes a training set of approximately 1.1 million reviews for over $11,000$ hotels, along with $1,050$ human-written summaries for 50 hotels. The dataset is designed to evaluate both general and aspect-specific opinion summarization models, with six popular aspects such as building, cleanliness, food, location, rooms, and service.

5 Evaluation Measures for Summarization

5.1 Automatic Evaluation Measures

ROUGE (Lin, 2004) The ROUGE score is a set of metrics used to evaluate the quality of automatic summarization and machine translation systems in natural language processing. It compares an automatically generated summary or translation with a reference or a set of reference summaries (typically human-produced). The ROUGE score ranges from 0 to 1, with higher scores indicating higher similarity between the generated summary and the reference. The most common ROUGE metrics to evaluate the summaries are:

1. ROUGE-1: This metric measures the overlap of unigrams (single words) between the system and reference summaries. It is defined as:

$$\text{ROUGE-1} = \frac{\sum_{i=1}^{|R|} \min(\text{C}(w_i, S), \text{C}(w_i, R))}{\sum_{i=1}^{|R|} \text{C}(w_i, R)}$$

where w_i is the i-th word in the reference summary R, S is the system-generated summary, and $\text{C}(w_i, X)$ is the number of times w_i appears in summary X.

2. ROUGE-2: This metric measures the overlap of bigrams (sequences of two words) between the system and reference summaries. It is defined as:

$$\text{ROUGE-2} = \frac{\sum_{i=1}^{|R|} \min(\text{C}(\text{bi}_i, S), \text{C}(\text{bi}_i, R))}{\sum_{i=1}^{|R|} \text{C}(\text{bi}_i, R)}$$

where bi_i is the i-th bigram in the reference summary R, and the counts are similar to those in ROUGE-1.

3. ROUGE-L: This metric measures the longest common subsequence (LCS) between the system and reference summaries. It is based on sentence-level structure similarity and identifies the longest co-occurring in sequence n-grams automatically. The ROUGE-L score is computed as:

$$\text{ROUGE-L} = \frac{LCS(R, S)}{|R|}$$

where $LCS(R, S)$ is the length of the longest common subsequence between the reference summary R and the system summary S, and $|R|$ is the length of the reference summary.

The ROUGE metrics provide a robust measure of the overlap between the generated summaries and the reference summaries, with ROUGE-1 and ROUGE-2 focusing on n-gram overlap and ROUGE-L emphasizing sequence similarity.

BLEU BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002), evaluates the similarity between candidate and reference summaries based on n-gram precision. It measures how many n-grams in the candidate summary match those in the ground truth summary. BLEU-1 assesses word-by-word matches, while BLEU-2 and higher consider matching pairs and longer sequences, respectively. Unigram scores gauge summary adequacy, indicating whether the model captures essential features, while higher n-grams assess fluency.

Despite its popularity in Natural Language Generation (NLG) systems, BLEU has limitations. Techniques like clipped precision address issues such as artificially inflated scores from repeated words, where each word is counted only up to its occurrence in the reference summary. Additionally, a brevity penalty discourages overly short summaries with mainly stop words, calculated based on the lengths of the predicted and reference sentences:

$$\text{Brevity Penalty} = \begin{cases} 1, & \text{if } c > r \\ e^{(1-\frac{r}{c})}, & \text{if } c \leq r \end{cases}$$

BERTScore (Zhang et al., 2019) BERTScore is a metric used to evaluate the quality of machine-generated text, particularly in tasks like summarization and machine translation. It leverages BERT (Bidirectional Encoder Representations from Transformers) embeddings to measure the similarity between the generated summary and the reference summary.

The BERTScore ranges from 0 to 1, with higher scores indicating better quality and greater similarity to the reference summary. The BERTScore metric consists of the following components:

1. BERT EMBEDDINGS: BERT embeddings are computed for both the generated summary S and the reference summary R.

2. COSINE SIMILARITY: The cosine similarity between the BERT embeddings of S and R is calculated to measure their similarity:

$$\text{Cosine Similarity} = \frac{\text{emb}(S) \cdot \text{emb}(R)}{\|\text{emb}(S)\| \cdot \|\text{emb}(R)\|}$$

   where $\text{emb}(X)$ represents the BERT embedding of summary X.

3. PRECISION: BERTScore also computes precision by comparing how well the generated summary captures important tokens from the reference summary:

$$\text{Precision} = \frac{\sum_{i \in R} \max_{j \in S} \text{emb}(r_i) \cdot \text{emb}(s_j)}{\sum_{i \in R} \text{emb}(r_i)}$$

   where r_i and s_j are tokens in R and S, respectively.

4. RECALL: BERTScore evaluates recall by measuring how well the reference summary tokens are captured by the generated summary:

$$\text{Recall} = \frac{\sum_{j \in S} \max_{i \in R} \text{emb}(r_i) \cdot \text{emb}(s_j)}{\sum_{j \in S} \text{emb}(s_j)}$$

5. F1 SCORE: The harmonic mean of precision and recall provides the overall BERTScore:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The BERTScore metric integrates BERT embeddings to assess both the content overlap and the quality of summary generation, making it a robust evaluation measure for tasks requiring semantic understanding and linguistic fluency.

5.2 Human Evaluation

Human evaluations, while more resource-intensive, provide critical insights into the quality of summarization that automatic metrics might overlook. Likert-scale evaluations (Likert, 1932) involve rating summaries on a numerical scale, typically from 1 to 5, based on criteria such as coherence, relevance, and readability. These scores are averaged across multiple human judges to obtain a reliable assessment of summary quality.

Best-Worst Scaling (BWS) (Louviere et al., 2015) is another human evaluation method that addresses some limitations of Likert scales. In BWS, evaluators are presented with sets of summaries and asked to identify the best and worst summaries based on specific criteria. This method reduces the cognitive load on evaluators and provides more discriminative results.

Comparative evaluations, including Win, Loss, and Tie assessments, involve directly comparing pairs of summaries. Evaluators determine which summary in each pair is better, worse, or if they are of equal quality. This method provides a relative evaluation that can be particularly useful when comparing different summarization models or approaches.

5.3 Large Language Model based Evaluation

The emergence of LLM-based evaluations represents a promising direction for summarization assessment. These evaluations utilize large language models to automatically assess the quality of generated summaries. For instance, GPT-3 and GPT-4 can be fine-tuned or prompted to evaluate summaries based on various criteria, such as coherence, relevance, and factual accuracy. These models can also generate detailed feedback, providing insights that go beyond simple scoring.

A recent study (Liu et al., 2023) has explored the use of LLMs for evaluation, leveraging their understanding of context and language to provide more accurate and nuanced assessments. Specifically, Liu et al. (2023) show that GPT-4 can be used as an evaluator in several tasks, including summarization. It shows great correlations with human-based evaluations of the tasks.

In summary, evaluation metrics for summarization encompass a range of methodologies, each with its strengths and limitations. Traditional reference-based metrics like ROUGE and BERTScore provide efficient and objective evaluations based on lexical and semantic similarities. Human evaluations offer critical insights into summary quality through direct human judgment, employing methods like Likert scales, Best-Worst Scaling, and comparative assessments. The advent of LLM-based evaluations introduces a powerful new tool for summarization assessment, combining the scalability of automatic methods with the nuanced understanding of human evaluations. Together, these metrics provide a comprehensive toolkit for evaluating the quality and effectiveness of summarization models.

6 Likelihood Maximization Approaches to Query-focused and Opinion Summarization

6.1 Query-focused Summarization

Various approaches have been proposed for QfS, ranging from extractive methods to sophisticated neural architectures.

Extractive Approaches. Wu et al. (2019) proposed an unsupervised extractive methodology using Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Blei and Lafferty, 2006) for topic modeling and pattern mining to score sentences. Sentences were selected based on these scores and a predefined compression ratio. This method alleviates the necessity for large datasets by leveraging unsupervised learning techniques. Mollá and Jones (2020) explored the domain of biomedical articles by utilizing a variety of machine learning and deep learning models for extractive QfS. They posed the task as a supervised learning problem where the model predicts the likelihood of a sentence being included in the final summary. Their experiments demonstrated the superiority of classification-based approaches over regression-based ones for this task. Additionally, Mollá and Jones (2020) highlighted that a Reinforcement Learning (RL) based training regimen, which trained a binary classifier to decide whether a sentence should be included in the summary, achieved better performance in human evaluations compared to supervised systems with similar ROUGE scores. The classifier utilized TF-IDF features of candidate sentences, the input text, the partially generated summary, remaining candidate sentences, the query, and the current length of the summary in sentences. The system was rewarded based on the ROUGE score of the generated summary.

Abstractive Approaches. Xu and Lapata (2021) addressed the challenge of data scarcity by generating proxy queries from generic summarization datasets using a Unified Masked Representation. This approach allowed for the training of an abstractive QfS model without the need for extensive QfS-specific datasets. They developed a mechanism that generates proxy queries to simulate the QfS task, enabling the use of existing large-scale generic summarization datasets for training. Laskar et al. (2020) leveraged transfer learning to improve QfS performance. They fine-tuned an abstractive summarizer, initially trained on the XSum dataset (Narayan et al., 2018), on the DebatePedia dataset. This method benefits from the robustness of models pre-trained on large datasets and further fine-tunes them on more specific QfS tasks, enhancing performance. Su et al. (2021) proposed enhancing QfS quality by incorporating an answer relevance representation into the decoder of a standard Transformer model. They employed a separate trained Question-Answering (QA) model to score each word's relevance in the context of the query, improving the relevance and coherence of the generated summaries. This method ensures that the generated summaries are more focused and directly answer the query.

6.2 Opinion Summarization

General Opinion Summarization. General opinion summarization aims to distill large sets of opinions into concise, coherent general summaries. Early methods focused on extractive techniques. For instance, Ganesan et al. (2010) leveraged redundancy in reviews to generate concise summaries, while Erkan and Radev (2004) used graph-based models to identify and select the most relevant sentences. More recent approaches have shifted towards neural network-based abstractive methods. Chu and Liu (2019); Bražinskas et al. (2020) use autoencoders (Kingma and Welling, 2013) and its variants to learn a review decoder through reconstruction which is then used to generate summaries using the averaged representations of input reviews. Another approach is to curate synthetic datasets using one of the reviews as a pseudo-summary and

pair it with input reviews using different strategies. Bražinskas et al. (2020) uses random sampling, Amplayo and Lapata (2020) generates noisy version of the pseudo-summary, Elsahar et al. (2021) ranks reviews using similarity and relevance, and Amplayo and Lapata (2020) uses content plans to generate synthetic datasets. Im et al. (2021) randomly selects a review as a pseudo-summary and proposes a pipeline to generate summaries using multimodal input such as text, image, and meta-data. Ke et al. (2022) captures the consistency of aspects and sentiment between reviews and summary, whereas Wang and Wan (2021) learns aspect and sentiment embeddings to generate relevant pairs. Iso et al. (2021) searches for convex combinations of latent vectors to generate summaries.

Aspect-specific Opinion Summarization. Aspect-specific opinion summarization focuses on generating summaries for specific aspects within reviews. Angelidis et al. (2021) proposed the first approach to generate both aspect-specific and general summaries. They utilize a Vector Quantized Variational Autoencoder (van den Oord et al., 2017) for clustering review sentences followed by a popularity-driven extraction algorithm to summarize. (Basu Roy Chowdhury et al., 2022) utilizes dictionary learning (Dumitrescu and Irofti, 2018) to acquire representations of texts based on latent semantic units. Amplayo et al. (2021) proposed the first abstractive approach for generating aspect-specific and general summaries. They generate synthetic datasets by identifying aspect-bearing elements (words, phrases, sentences) using a multiple instance learning (MIL) (Keeler and Rumelhart, 1991) model trained on silver-labeled data obtained through seed words. Shen et al. (2023) proposes two simple solutions for generating synthetic datasets that do not rely on complex MIL modules. The SW-LOO simply matches the aspect seed words to construct synthetic datasets, whereas NLI-LOO uses an off-the-shelf NLI model to do so using only aspects and no seed words. Mukherjee et al. (2020) takes an unsupervised approach to extract aspects and manually create a mapping between fine-grained and coarse-grained aspects using Integer Linear Programming (ILP) based extractive subset of opinions.

Self-Supervised Opinion Summarization. Recent approaches use self-supervision by considering one of the reviews as a pseudo-summary. Bražinskas et al. (2020) randomly selected N reviews per entity to construct N pseudo-summary, reviews pairs. Amplayo and Lapata (2020) sampled a review randomly and generated noisy versions of it as input reviews. Amplayo et al. (2020) used aspect and sentiment distributions to sample pseudo-summaries. Elsahar et al. (2021) selected reviews similar to a randomly sampled pseudo-summary as input reviews, based on TF-IDF cosine similarity. Wang and Wan (2021) aimed at reducing opinion redundancy and constructed highly relevant reviews pseudo-summary pairs by learning aspect and sentiment embeddings to generate relevant pairs. Im et al. (2021) used synthetic dataset creation strategy similar to Bražinskas et al. (2020) and extended it to multimodal version. Ke et al. (2022) captured the consistency of aspects and sentiment between reviews and pseudo-summary using constrained sampling. Siledar et al. (2023a) use lexical and semantic similarities for creating synthetic datasets.

Multi-Source Opinion Summarization. Multi-source summarization integrates information from various sources to generate more comprehensive summaries. Zhao and Chaturvedi (2020) used aspects identified from product description to perform extractive aspect-based opinion summarization. Li et al. (2020) proposed a supervised multimodal summarization model to effectively generate summaries using reviews, product image, product title, and product details. Im et al. (2021) proposed a self-supervised multimodal training pipeline to generate summaries using reviews, images, and meta-data. Siledar et al. (2023b) did supervised opinion summarization using simple rules to generate summaries separately in the form of verdict, pros, cons, and additional information using reviews, description, specifications, and question-answers.

7 Reward-based Approaches to Query-focused and Opinion Summarization

Supervised Learning has been frequently used for several tasks that involve Text Generation (Nallapati et al., 2016; Nema et al., 2017; Siledar et al., 2023c). We have seen a few works using Supervised Learning specifically for Queruy-focused Summarization and Opinion Summarization in Section 6. However, for such tasks, Supervised learning suffers from several crucial problems. Two of

the most prominent ones are listed below:

- ***Exposure Bias*:** Traditional Sequence-to-Sequence models are trained using Teacher Forcing (Williams and Zipser, 1989). This enables faster training, which requires less memory, by reducing the length of the gradient chain in the Back Propagation Through Time (BPTT) algorithm. Teacher Forcing refers to the usage of ground truth tokens (y_1, y_2, $\cdots$, y_{t-1}) for generating the y_t token during training. Such a formulation helps in parallel generation of tokens of all time-steps, thereby reducing the time of training. However, such a training leads to a bias during inference time within the model: "*whatever has been thus-far is good enough, there is no need to rethink*". This bias is known as **exposure bias**.

- ***Train/Test Mismatch*:** Supervised Learning uses Cross-Entropy loss to train the Text Generation models. It does not easily allow any task-specific properties into the training objective. This leads to a mismatch between the true objective (promoting task-specific properties) and the train objective (maximizing likelihood of the given dataset).

These problems within Supervised Learning (especially the second one) can be mitigated using Reinforcement Learning. This motivation has been used by several works in contribution to the Reinforcement Learning for Text Generation literature. In this section, we will look at a few such representative works; however, we will be concentrated on Reinforcement Learning for Text Summarization, as that is more relevant to our survey.

We find an abundance of precedence of Reinforcement Learning (RL) for abstractive summarization. Paulus et al. (2017) provide first work in employing RL to train LSTM models, without Teacher Forcing, for abstractive summarization. The authors employ a mixed-objective loss (cross-entropy loss + policy gradient loss). This was done because cross-entropy loss objective kept the model grounded to the language properties, while the policy gradient loss objective helped promote the summarization specific properties within the model. The authors used ROUGE as the only reward in the Reinforcement Learning framework. Pasunuru and Bansal (2018); Li et al. (2019) provide follow-up works with better reward functions. Both these works stress that ROUGE *is not a sufficiently good reward for Summarization*, as it does not promote properties like semantic similarity, or penalize properties like redundancy within the generated summary. Pasunuru and Bansal (2018) provide an additional reward Saliency that checks if the summary incorporates important aspects. Li et al. (2019) propose the usage of BERTSCORE as an additional reward, to promote semantic similarity. In a similar way, Roit et al. (2023); Tang et al. (2023) attempted to reward factuality within generated summaries. Specifically, Roit et al. (2023) used Natural Language Inference (*does the document entail the summary?*) to promote factuality within generated summaries. Tang et al. (2023) attempt using a similar factuality metric to promote grounded summary generation in the Clinical Text domain. Recently, Nath et al. (2023) have proposed a novel reward mechanism based on a new Passage Embedding approach, which promotes better semantic similarity with the ground truth summaries-. All of these works have been faithful to the following theme: "*Finding better rewards for the Task*".

Recently Reinforcement Learning from Human Feedback (RLHF; Ziegler et al. (2019); Bai et al. (2022); Ouyang et al. (2022); Rafailov et al. (2023)) has emerged as a new paradigm in applying Reinforcement Learning to Natural Language Generation. In this approach, a separate Reward Model is learned to grant rewards to generated text in the Reinforcement Learning training pipeline. This reward model is learned using human preference data (preference over which text humans prefer for a given input). A big challenge, however, in RLHF is that the reward model needs additional data for training (typically tens of thousands in size; Nakano et al. (2021); Bai et al. (2022); Ethayarajh et al. (2022)). Recently, Nath et al. (2024) provide a way to mitigate such a requirement. Specifically, Nath et al. (2024) demonstrate this in the domain of Opinion Summarization, where they achieve a reduction in dataset requirement by $21\times$.

8 Open Challenges and Future Reserach Directions

While there has been significant growth in employing Reinforcement Learning for Text Summarization, these approaches lack discipline in choosing the rewards. There is no governing principle which states the necessary and sufficient set of rewards for a tasks. Reinforcement Learning from Human

Feedback has attempted to mitigate this by directly aligning with human goals. It has, to some extent, mitigated the necessity of such a governing principle. Recently, Nath et al. (2024) attempt at arriving a necessary set of rewards, further showing how these rewards correlate with human notion of a good Opinion Summary. Developing such a governing framework has two fold benefits:

- It provides a trust on the set of rewards used for the task—since the rewards are arrived at through the framework, they are trivially trustworthy.
- It provides an evaluation framework for techniques such as RLHF—how does a model trained using RLHF fare on these reward metrics?

9 Summary and Conclusion

In this comprehensive we have covered several works that have advanced the state-of-the-art in research on Query-focused Summarization and Opinion Summarization, through the lens of Reinforcement. We provide a view on how the works shifted their focus from Supervised Learning to Reinforcement Learning, gradually arriving at the conclusion that designing task-aligned rewards leads to better performance. We have seen how simpler Reinforcement Learning approaches with ROUGE as the reward were gradually replaced by more sophisticated rewards, such as BERTSCORE, Passage Embedding, etc. Finally, we have concluded this survey by providing a long-term future research direction which can significantly impact the state of Reinforcement Learning in Natural Language Generation.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2020. Unsupervised opinion summarization with content planning. In *AAAI Conference on Artificial Intelligence*.

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945, Online. Association for Computational Linguistics.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Anthropic. 2023. Claude: An ai assistant by anthropic. `https://www.anthropic.com/claude`. Available at `https://www.anthropic.com/claude`.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, T. J. Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, abs/2204.05862.

Somnath Basu Roy Chowdhury, Chao Zhao, and Snigdha Chaturvedi. 2022. Unsupervised extractive opinion summarization using sparse coding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1225, Dublin, Ireland. Association for Computational Linguistics.

Tal Baumel, Raphael Cohen, and Michael Elhadad. 2016. Topic concentration in query focused summarization datasets. In *AAAI Conference on Artificial Intelligence*.

Manik Bhandari, Pranav Gour, Pengfei Liu, and Zhe Fu. 2020. Re-evaluating evaluation in text summarization. *arXiv preprint arXiv:2004.06676*.

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 113–120, New York, NY, USA. Association for Computing Machinery.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2021. Learning opinion summarizers by selecting informative reviews. In *Proceedings of the 2021 Conference*

on Empirical Methods in Natural Language Processing, pages 9424–9442, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

Hoa Trang Dang. 2005. Overview of duc 2005. In *In Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP*.

Hoa Trang Dang. 2006. DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55, Sydney, Australia. Association for Computational Linguistics.

Google DeepMind. 2023. Gemini: A language model by google deepmind. https://www.deepmind.com/research/gemini. Available at https://www.deepmind.com/research/gemini.

Bogdan Dumitrescu and Paul Irofti. 2018. *Dictionary learning algorithms and applications*. Springer.

Hady Elsahar, Maximin Coavoux, Jos Rozen, and Matthias Gallé. 2021. Self-supervised and controlled multi-document opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1646–1662, Online. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of ACL 2019*.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China. Coling 2010 Organizing Committee.

Jinbae Im, Moonki Kim, Hoyeop Lee, Hyunsouk Cho, and Sehee Chung. 2021. Self-supervised multimodal opinion summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 388–403, Online. Association for Computational Linguistics.

Hayate Iso, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, and Wang-Chiew Tan. 2021. Convex Aggregation for Opinion Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3885–3903, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenjun Ke, Jinhua Gao, Huawei Shen, and Xueqi Cheng. 2022. Consistsum: Unsupervised opinion summarization with the consistency of aspect, sentiment and semantic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 467–475.

Jim Keeler and David Rumelhart. 1991. A self-organizing integrated segmentation and recognition neural net. *Advances in neural information processing systems*, 4.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Huang. 2020. Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models. In *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings*, page 342–348, Berlin, Heidelberg. Springer-Verlag.

Haoran Li, Peng Yuan, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Aspect-aware multimodal summarization for chinese e-commerce products. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8188–8195.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages

6038–6044, Hong Kong, China. Association for Computational Linguistics.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. 2015. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.

Diego Mollá and Christopher Jones. 2020. Classification betters regression in query-based multi-document summarisation techniques for question answering. In *Machine Learning and Knowledge Discovery in Databases*, pages 624–635, Cham. Springer International Publishing.

Rajdeep Mukherjee, Hari Chandana Peruri, Uppada Vishnu, Pawan Goyal, Sourangshu Bhattacharya, and Niloy Ganguly. 2020. Read what you need: Controllable aspect-based opinion summarization of tourist reviews. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1825–1828.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Ouyang Long, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv*, abs/2112.09332.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *CoRR*, abs/1808.08745.

Swaroop Nath, Pushpak Bhattacharyya, and Harshad Khadilkar. 2023. Reinforcement replaces supervision: Query focused summarization using deep reinforcement learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15770–15789, Singapore. Association for Computational Linguistics.

Swaroop Nath, Tejpalsingh Siledar, Sankara Sri Raghava Ravindra Muddu, Rupasai Rangaraju, Harshad Khadilkar, Pushpak Bhattacharyya, Suman Banerjee, Amey Patil, Sudhanshu Shekhar Singh, Muthusamy Chelliah, and Nikesh Garera. 2024. Leveraging domain knowledge for efficient reward modelling in rlhf: A case-study in e-commerce opinion summarization.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada. Association for Computational Linguistics.

OpenAI. 2023. Chatgpt: Language model developed by openai. `https://www.openai.com/chatgpt`. Available at `https://www.openai.com/chatgpt`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653, New Orleans, Louisiana. Association for Computational Linguistics.

Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data augmentation for abstractive query-focused multi-document summarization. In *AAAI*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2523–2544, Online. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Leonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos Garea, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6252–6272, Toronto, Canada. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Ming Shen, Jie Ma, Shuai Wang, Yogarshi Vyas, Kalpit Dixit, Miguel Ballesteros, and Yassine Benajiba. 2023. Simple yet effective synthetic dataset construction for unsupervised opinion summarization. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1898–1911, Dubrovnik, Croatia. Association for Computational Linguistics.

Tejpalsingh Siledar, Suman Banerjee, Amey Patil, Sudhanshu Singh, Muthusamy Chelliah, Nikesh Garera, and Pushpak Bhattacharyya. 2023a. Synthesize, if you do not have: Effective synthetic dataset creation strategies for self-supervised opinion summarization in E-commerce. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13480–13491, Singapore. Association for Computational Linguistics.

Tejpalsingh Siledar, Jigar Makwana, and Pushpak Bhattacharyya. 2023b. Aspect-sentiment-based opinion summarization using multiple information sources. In *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, pages 55–61.

Tejpalsingh Siledar, Jigar Makwana, and Pushpak Bhattacharyya. 2023c. Aspect-sentiment-based opinion summarization using multiple information sources. In *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD), Mumbai, India, January 4-7, 2023*, pages 55–61. ACM.

Tejpalsingh Siledar, Rupasai Rangaraju, Sankara Sri Raghava Ravindra Muddu, Suman Banerjee, Amey Patil, Sudhanshu Shekhar Singh, Muthusamy Chelliah, Nikesh Garera, Swaprava Nath, and Pushpak Bhattacharyya. 2024. Product description and qa assisted self-supervised opinion summarization.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. 2021. Reward is enough. *Artificial Intelligence*, 299:103535.

Dan Su, Tiezheng Yu, and Pascale Fung. 2021. Improve query focused abstractive summarization by incorporating answer relevance. In *FINDINGS*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, second edition. The MIT Press.

Xiangru Tang, Arman Cohan, and Mark Gerstein. 2023. Aligning factual consistency for clinical studies summarization through reinforcement learning. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 48–58, Toronto, Canada. Association for Computational Linguistics.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ke Wang and Xiaojun Wan. 2021. TransSum: Translating aspect and sentiment embeddings for self-supervised opinion summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 729–742, Online. Association for Computational Linguistics.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Yutong Wu, Yuefeng Li, and Yue Xu. 2019. Dual pattern-enhanced representations model for query-focused multi-document summarisation. *Knowl. Based Syst.*, 163:736–748.

Yumo Xu and Mirella Lapata. 2021. Generating query focused summaries from query-free resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6096–6109, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. *arXiv preprint arXiv:1904.09675*.

Chao Zhao and Snigdha Chaturvedi. 2020. Weakly-supervised opinion summarization by leveraging external information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9644–9651.

Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593.