SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Reinforcement Learning Approaches for Faithful Abstractive Text Summarization

Jakob Sturm

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Reinforcement Learning Approaches for Faithful Abstractive Text Summarization

Reinforcement Learning Ansätze für ursprungsgetreues abstraktives Textzusammenfassen

Author:	Jakob Sturm
Supervisor:	Prof. Dr. Georg Groh
Advisor:	M.Sc. Tobias Eder
Submission Date:	16.08.2023

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 16.08.2023 Jakob Sturm

Abstract

The amount of available data is rapidly growing. Filtering out the relevant information is especially crucial in domains like political news. Automatic summarization is one potential tool to tackle this task. However the current state-of-the-art is still challenged to overcome some obstacles. One particular issue is the generation of unfaithful summaries. Reinforcement Learning (RL) is considered a potential solution, as it circumvents the need for gold label references and other issues of supervised approaches.

This thesis works out the importance of faithfulness and its relation to overall summary quality. Different RL approaches are presented and their strengths and weaknesses are discussed. The current focus on reward design is highlighted. Approaches based on learning from human feedback are potentially well aligned but resource intense. Learning based on metrics is accessible but prone to overfitting.

A diverse set of reference-free metrics, as potential reward building blocks, is evaluated using the recent AGGREFACT benchmark: Grusky, Naive, ROUGE, DAE, QuestEval, SummaC, DiscoScore, ESTIME, SGwLM, BLANC, Perplexity, BERTScore, CTC and ChatGPT-ZS, -CoT, -DA and -Star. An intriguing finding is the good performance of simple metrics based on word matches, which are nearly on par with the best evaluated recent metric. This casts doubt either on the improvement of recent metrics or the distinguishing power of AGGREFACT.

For each selected metric we evaluate available sub-versions and determine which is the best on CNN/DM- and XSUM-like data. We find that performance on one is not linked to performance on the other. Additionally, we find that the recent ChatGPT-CoT metric is heavily influenced by the selected heuristic component.

As rewards are often composed of different metrics, to combine multiple parts into one overall quality estimation, and cover weakness of individual metrics, we propose to learn such a combination. We train an ensemble, utilising a diverse set of metrics as features, based on a Neural Network architecture, for overall quality. The hypothesis is that improvement for overall quality will eventually also improve faithfulness. The training is successfully performed utilizing an Evolutionary Algorithm. However gains do not translate to faithfulness. Performance on CNN/DM remains about as high as the top features, but performance on XSUM drastically diminishes. A further analysis shows that our training data is more aligned to the first data set.

In sum, we provide an overview of RL approaches and metrics, and support with our analysis the current call to collect more human judgment data to reliably evaluate and train metrics.

Kurzfassung

Die Menge der verfügbaren Daten nimmt mit steigender Geschwindigkeit immer weiter zu. Die relevanten Informationen zu finden ist besonders wichtig in Feldern wie Nachrichten zu politischen Themen. Ein Ansatz, um dieses Problem in den Griff zu bekommen, ist automatisches Textzusammenfassen. Jedoch stellen sich dem aktuellen Stand der Technik noch einige Herausforderungen. Besonders die Tatsache, dass Inhalte von Zusammenfassungen nicht immer zum Ursprunstext getreu sind, ist ein Problem. RL wird als ein möglicher Lösungsansatz gesehen, da es die Abhängigkeit von Musterzusammenfassungen und andere Probleme von Supervised Learning umgeht.

Diese Arbeit erläutert die Bedeutung der Urspungstreu von Zusammenfassungen und die Beziehung dieser Eigenschaft zu allgemeiner Qualität. Unterschiedliche RL-Ansätze werden vorgestellt, wobei Vor- und Nachteile herausgestellt und diskutiert werden. Der aktuelle Focus auf das entwickeln von Reward-Funktionen wird betont. Ansätze, die direkt von menschlichen Einschätzungen lernen, erfassen implizite Vorstellungen von Qualität potentiell sehr gut, sind dafür aber sehr resourcenaufwendig. Lernen, das auf Metriken basiert, ist leichter zugänglich, kann aber zu ungewünschten Ergebnissen führen, dadurch das Schwächen in der Reward-Funktion ausgenutzt werden.

Es wird ein vielfältiger Satz an referenzfreien Metriken, die als Reward-Bausteine dienen können, evaluiert: Grusky, Naive, ROUGE, DAE, QuestEval, SummaC, DiscoScore, ESTIME, SGwLM, BLANC, Perplexity, BERTScore, CTC and ChatGPT-ZS, -CoT, -DA and -Star. AGGREFACT ist der hierzu verwendete Test-Datensatz und Vorgang. Eine verblüffende Entdeckung ist das gute Abschneiden von einigen einfachen Metriken, die auf Wortübereinstimmungen basieren. Diese sind beinahe gleich auf mit den besten der aktuelleren Metriken. Das wirft entweder Zweifel auf den Fortschritt neuer Metriken oder auf die Fähigkeit von AGGREFACT das Potenzial von Metriken zu unterscheiden.

Für jede der ausgewählten Metriken untersuchen wir unterschiedliche Varianten und bestimmen die beste bezüglich jeweils CNN/DM-ähnlichen und XSUM-ähnlichen Daten. Wir finden heraus, dass das Abschneiden auf dem Einen nichts über das Abschneiden auf dem Anderen aussagt. Außerdem beobachten wir, dass die kürzlich vorgestellte Metrik ChatGPT-CoT sehr abhängig davon ist, welche Heuristik als Komponente verwendet wird.

Da Reward-Funktionen oft aus unterschiedlichen Metriken zusammengesetzt sind, um einzelne Aspekte zu einer Einschätzung für allgemeine Qualität zu kombinieren und Schwächen einzelner Metriken auszugleichen, stellen wir einen Ansatz vor, solch eine Kombination zu lernen. Wir trainieren ein Ensemble, das sich aus mehreren Metriken als Eingaben speist, mit Hinblick auf allgemeine Zusammenfassungsqualität. Die verwendete Architektur basiert auf einem Künstlichen-Neuralen-Netzwerk. Die Hypothese ist, dass Verbesserung der allgemeinen Qualität auf lange Sicht eine Verbesserung der Urspungsgetreue mit sich bringt.

Das Lernen konnte basierend auf einem Evolutionären-Algorithmus erfolgreich durchgeführt werden. Jedoch überträgt sich der Fortschritt nicht. Das Ensemble schneidet im Test auf CNN/DM nur in etwa so gut ab wie seine besten Bestandteile. Auf XSUM bricht die Leistung sogar deutlich ein. Eine darauf folgende Analyse ergibt, dass der Trainingsdatensatz wesentlich ähnlicher zu CNN/DM als zu XSUM ist.

Zusammenfassend stellen wir eine Übersicht über Reinforcement Learning Ansätze und eine über Metriken vor. Außerdem unterstützen unsere Analysen den aktuellen Ruf nach mehr Daten mit menschlichen Bewertungen der Qualität von Zusammenfassungen. Diese sind nötig, um Metriken zuverlässig entwickeln und bewerten zu können.

Contents

1. Introduction

The amount of available information is growing. Keeping track is getting more and more challenging. This is problematic for domains like science or especially politics. In a democratic society, there is a significant responsibility on every individual to exercise their right to vote based on an informed foundation. One approach to cope with the information flood is Automatic Text Summarization (ATS).

Summarization is the process of transforming a given text into a shorter one while keeping the important information. That the resulting summary should be comprehensible is a given. [1, 2]

Automating this process and so freeing up human labour capacity came into reach with the enhanced powers of NLP and large pretrained language models like GPT-3 [3].

Deployment of this tool can be seen in the previews provided by search engines [4]. Impressive capabilities can be experienced by everybody using ChatGPT.

Despite this enhancement recent summarizers are still flawed. A particular issue is the creation of unfaithful content, statements that are not backed or even contradicted by the original document [5, 6, 7, 8, 9].

Unaccomplished qualities like linguistic fluency can make a summary useless, but unfaithful content can be harmful. If the summary is just bad the reader will switch to the original document, if the summary looks good but is unfaithful the reader might make decisions based on wrong information. The lack of faithfulness is considered a major problem for the wider application of ATS. [10, 7, 11, 5, 12]

To guide the research community on their way to create better summarizers, good estimation of system- and summary-quality is important. That is a challenge [13, 3] as summary quality is only vaguely defined [14, 15]. Imperfect judgment makes spotting improvements and weaknesses very hard, can even be misleading [16, 17, 18]. Humans examining summary quality is considered the gold standard [19, 20, 13, 21, 22], but is so time consuming that it can hardly be done in the required scope [13, 15]. Automatic metrics, developed to improve the accessibility of quality assessment [14], are therefore utilized in numerous papers.

However, developing a reliable metric is an open challenge [14, 15, 23, 18, 24, 25, 26, 13], and also evaluating the quality of metrics faces difficulties [15, 27]. Consequently, the community lacks certainty about which metric is the best, and even the elaborated ones have known flaws [28, 29, 16, 15, 30, 5]. Also building trust by relying on multiple metrics is an unstable approach as metrics do not agree with another [24, 25]. Additionally, many papers still rely on older metrics like ROUGE [31] [18, 4, 32, 33, 20], despite their widely discussed limitations [34, 4, 30, 35, 36], primarily due to their established use [18].

This caveats apply to metrics in the context of overall quality as well as faithfulness.

Very tightly tied to the topic of summary quality evaluation is the approach of Reinforcement Learning (RL) for summarization. In contrast to supervised summerizer training, RL needs continuous summary quality assessment during the training process, amplifying the assessments influence. Despite this caveat [15], RL is seen as a promising research direction as it could solve many drawbacks of the supervised approach [37, 20, 13, 38]. e.g the problem of exposure bias [39, 42, 43, 19, 44, 45, 37, 14, 20, 40, 41]. Furthermore RL is able to evaluate the summary as a whole [46, 14, 38], which, compared to the token level assessment of supervised learning, is closer to the final evaluation.

This thesis provides a high level classification of Reinforcement Learning for summarizer training and an evaluation of some metrics, ranging from established baselines to SOTA. Furthermore an ensemble of metrics is trained and analysed.

The overview emphasises the current focus on reward development and presents different RL approaches using reward as the main distinguishing characteristic. Due to the importance of metrics for evaluation and their common use as building blocks for rewards, we try to make a contribution in this realm. To decide which metric has the most potential as building block, we update a recent meta evaluation, based on the AGGREFACT benchmark, [47] with additional metrics. As rewards often are made up of multiple parts, and inspired by the NUBIA paper [48], we evaluate whether a trained ensemble of metrics can yield an improvement on the benchmark. Furthermore, we introduce two approaches to enhance recent ChatGPT-based metrics and analyse their effect using AGGREFACT.

Relevant code utilized to perform our experiments and analysis is provided on Github.[1]

1.1. faithfulness and overall quality

A summary is faithful if every contained information piece is supported by the original document. Made up information is called hallucination. It is important to stress that a hallucination does not need to be factually incorrect. Faithfulness only describes the relation of a summary to the original document not to any world knowledge. [19, 23, 20] Factuality [49] is a quality aspect on its own.

While faithfulness is very important, there is no benefit in optimizing it in isolation. An empty summary can not contain any hallucinations. Faithfulness in isolation is trivial. It needs to be combined with other quality aspects, like: Readability [42, 50, 51, 52, 53, 13, 54], Fluency [55, 56, 57, 58, 49, 53, 13], Semantic-coherence [34, 56, 59, 58, 49, 45, 53], Relevance [42, 34, 56, 50, 58, 49, 53], Redundancy [50, 52, 53, 54], Informativeness [53, 13], Length [57], Conciseness [34, 51, 60, 13] and Coverage [51, 45, 26], to be meaningful.

How to combine this aspects [61, 33, 4, 1, 49, 45] to meet human expectation is an open task [14, 1, 62, 45], but overall quality can be directly assessed by humans. Aggregating metrics, tailored for different aspects, into one reward is significantly more of an issue than combining them for evaluation. While evaluation is typically performed only once, human interpretation,

[1] `https://github.com/UntotaufUrlaub/RL_for_summarization_based_on_faithful_metrics`

aggregating different aspects, is well-suited for this task. However, as Reinforcement Learning requires multiple reward calculations, human involvement might become prohibitively expensive. Nonetheless, aggregation remains necessary as Reinforcement Learning relies on a single reward value for each action.

Our evaluations are performed with regard to faithfulness, because we think being faithful is an important first step to being overall good. For the described reasons our ensemble training is performed with respect to overall summary quality. As we see this as potential reward used in RL, isolated faithfulness is of no use. Therefore we check if the ensemble can improve on the overall quality validation set in section 6.5 and whether this improvement translates to faithfulness (section 7.3). Furthermore we report the performance of the metrics on our training data set and compare it to the faithfulness results in section . This evaluation is not as elaborated as the benchmark and so should only be seen as an additional hint on how to value the metrics.

1.2. Thesis Outline

Section 2 outlines fundamental concepts, including the assessment of summary quality and metrics. It discusses various classes of metrics and their relevant concepts, addresses the overfitting issue in the realm of RL, and elucidates the training process employed for our ensemble.

In Section 3, we present and discuss various Reinforcement Learning approaches. The metrics introduced in Section 4 are used as features for the ensemble, as described in Section 6, and are evaluated in Section 7. We introduce AGGREFACT as the selected benchmark for our metric assessment in Section 5. Our conclusion and potential directions for future work are outlined in Sections 9 and 8 respectively.

We deem that the following two points are our most relevant research contributions:

- The dependence of the ChatGPT-CoT metric onto an elaborated heuristic. (Section 7.2.12)
- The fact that simple metrics, based on word matching, are nearly as good as recent metrics on AGGREFACT. (Section 7.3)

While Training of the ensemble was performed successfully, the gain did not translate to faithfulness. We assume the reason lies in insufficient training data and support this by an analysis in section 7.4.

2. Material

2.1. Quality estimation approaches

Human judgment and metrics are both frequently used to assess summary quality [20], either overall or for individual aspects. Each approach has its advantages and disadvantages. For evaluation it is common practice to pair them [13, 4].

2.1.1. Human evaluation

Human evaluation is considered the gold standard [19, 20, 13, 21, 52] but has to be carefully designed and executed [1, 63, 22], as it is considered to be difficult [20, 26, 64, 13, 61]. Challenges are the subjective nature of human ratings [1, 63, 34, 26, 30, 61], that there is no consensus for the design approach yet [65, 26, 22, 1, 30], and the immense amount of needed labour time [20, 21, 52, 25, 14, 13].

Ratings by individual humans are influenced by the persons background, whether they are experts or laymen [66], whether the target language is their first or second and their level of education [13].

Some papers [66, 67, 68] argue that structuring and decomposing the task [52, 2, 22, 26] is beneficial. Others [30, 69] suspect bias to be introduced by any advanced task design.

Utilizing absolute or relative quality ratings is one particular design decision [26].

For the first approach each summary is evaluated individually. The rating is performed on a scale, the granularity of which can vary. Typically a lower values means worse and a higher value means better. For example five star system (e.g. see LIKERT scale in [52]) or zero to 100 percent are common.

The second approach relies on comparing summaries to another, deciding for better and worse. Comparing two summaries or three, which is then referred to as Best-Worst-Rating, is common [70, 71]. The papers [72, 73, 71] argue that relative ratings produce more reliable results. For interpretation of a mass of relative ratings they are usually transferred into a scale like in these papers: [72, 71]. The paper [74] by Hollis discusses different transformation schemes. We will later rely on ELO as they describe it, because of the familiarity, from realms like chess, and the good performance in their analysis. Further more it is simple and has a good time-complexity. Though it might be suboptimal if the data is too noisy. [74]

To perform the ELO calculation ratings have to be converted to binary matches. For binary ratings this is trivial, the better summary is considered the winner. If three summaries are compared, they are rated as worse, medium and better. Here we created two matches, better wins against medium, and medium against worse.

Transformation is performed from an example level (see 2.2) perspective, which means summaries are the participants in the matches. If we would investigate the system level performance summarizers would the participants. Consequently the first approach yields an overall ranking of summaries and the second would yield a ranking of summarizers.

The resulting rankings can split into separate rankings if the graph of matches (summaries as nodes and matches as edges) is not fully connected. Each fully connected sub-graph produces its own scale. If no elements of two sub-graphs are compared they can not be placed on a unified ranking. Their individual rankings might have different scales and offsets.

We want to stress that the ELO calculation is to some degree dependent on the order of matches. As none is given for summary quality rankings, we assumed random order. To mitigate the effect of this randomness, we averaged the results of multiple ELO calculations.

2.1.2. Metrics

The development of metrics is hindered by the problems human evaluation faces. Time demand hampers creation of big data sets, which could be used for training or evaluation. Disagreement in the protocol selection prevents created data sets to be universally established.

Metric development aims to overcome the significant barriers of time and cost associated with human evaluation. While this requirement can already be met, the level of validity might not be comparable at the moment. Consequently, both approaches need to be selected cautiously and combined when necessary.

Numerous designs have emerged during the continuous search for metrics at a satisfactory level. Thus, we present an overview of various types of approaches.

Metric Classes

One possible distinction is the kind of data used during metric creation. Options are: human generated data, synthetic data and no data.

Supervised training on human generated data is believed to be a potent approach [24], but as described in 2.1.1 sufficient amounts of human generated data are hardly available [49].

To circumvent the lack of training material, often synthetic data is created. Heuristics are used to expand a small amount to a full training set. This approach is called semi-supervised or weakly supervised training [19, 75, 76, 77, 78]. A realistic representation of different quality levels and errors is the key to enable successful metric development, but that requirement seems hardly met [28, 79, 15, 5, 80, 12]. [24] proposes a combination: semi-supervised pre-training, followed by supervised fine-tuning.

No data at all is needed when the metric is not trained but instead follows some handcrafted logic's.

Metrics can also be classified into QA-, NLI-, classifier-, perplexity-, IE-, embeddings- and word-based, which describe the most popular building concepts. [81, 82, 15, 29]

Traditional metrics relied on analysing words, word-parts or word-groups. Frequencies are calculated for the summary and either the original document or gold label reference

summaries. Comparison allows estimation of quality by the proxy of similarity. Members are simple approaches like comparing the frequency of 'any word', to make a claim based on length relations, and the famous ROUGE-score [31]. [18, 4, 32, 33, 20] The lacking ability of word-based metrics to consider synonyms and deeper meaning [33] is cause for criticism and the development of other approaches.

A proposed solution is calculating relations based on learnt embeddings of words (tokens), sentences or even whole texts, as they handle synonyms and capture some degree of context and meaning [83, 29, 15].

The Question Answering (QA) -based approaches states that, if one text is a stand-in for another, both should allow answering the same questions with similar answers. A good summary should be a proxy for the original text, or should be close, in terms of content, to some reference summaries. Questions are generated using (DL-based) question extraction modules based on the text or the references. The answers, delivered by (DL-based) question-answering modules, are compared, using varying approaches, to calculate a final score. [15] While this is a common approach [84], the three-step structure is also seen problematic. Discussed issues are cascading errors from one module to the next and high computational requirements [75]. NLI (see 2.1.2) is therefore often seen as more promising [84, 75].

Decomposing each text into small information pieces, called Information Extraction (IE), and matching them is especially tailored to content comparison, as linguistic and stylistic aspects are stripped away. [85]

Natural Language Inference (also known as Textual Entailment) (NLI) is a self-reliant research area, as is question-answering and question-extraction. NLI-based metrics typically do not rely on composing NLI modules, but have repurposed them. The assumption is that two texts, which are similar in terms of content, may not look similar, but won't contradict or even better entail each other. [86, 7, 87, 88, 89, 90, 91, 23, 92]

We title the approach of directly learning a final score from quality judgment data as classifier-based. Of course the other approaches may involve some training as well, but that is typically restricted to the realm of fine-tuning. On the other hand, classifiers can also make use of pretrained features. The distinction is the assigned importance between components and training. [15, 52, 49, 76] Classifiers are seen as very potent and flexible but are held back by the lack of high quality training data [24].

Several metrics, categorized as based on token-likelihood, utilize probabilities derived from Large Language Models (LLMs). For instance, good perplexity of a text is associated with high linguistic quality [93]. Another approach involves analyzing the effect of knowing one text on the perplexity of another. A well-matched summary should have higher likelihood when the text is given. The original text should be more likely given the summary, as the main information is already familiar. [19, 94, 95]

Whether the metric is trained or untrained is another categorization dimension. Trained metrics like classifiers rely on human generated quality estimation data or synthetic replacements. Logic-based metrics, on the other hand, can be created without the need for training, although they may involve pretrained building blocks. Examples in this category are ROUGE and HaRiM+ [95]. Logic based approaches are sometimes further distinguished into untrained

and hybrid, with the later describing the use of pretrained features. [24, 96, 78]

Some metrics need a reference or better a set of high quality references to calculate their score. Judgment is performed based on similarity hoping the embedded qualities in the references are taken into account. Reference free metrics only inspect a text in isolation, to assess for example linguistic qualities, or take a summary and the original text to compare desired features like information content. Reference free metrics might need more elaborated design, but eliminate the burden of collecting references, which is often seen as a bottle neck. [97, 5, 79]

The paper [97] proposes repurposing reference-based metrics in a reference-free manner. Since the required gold labels are often of poor quality [5, 80, 98, 18, 99, 100] and similarity should also be given between the original text and the summary, they replace the reference with the original text. Motivated by the promising results of their analysis, we adopt this approach. Other papers, like [95], also rely on this procedure. Additionally, while still under discussion, some papers argue that the reference free approach might be superior to the reference based one in the context of summarization [44, 15, 26, 9].

Usually metrics are designed to be deterministic, but there are exceptions. Building blocks might cause randomness, as is the case for the ChatGPT-based metrics. We refer to the later category as stochastic metrics.

2.2. Meta evaluation for metrics

To determine the performance of competing metrics is important. Collecting a data set of text, summaries and associated human quality assessment is the established procedure. Metric performance can either be assessed based on accuracy or correlation. Accuracy is used by AGGREFACT [47] and others [79, 101, 94, 75, 76, 80, 5, 79]. Usage of correlation can, for example, be seen in any of these papers: [79, 55, 44, 106, 56, 107, 2, 94, 75, 92, 102, 58, 100, 49, 29, 45, 52, 103, 48, 104, 105, 95].

While this are the most common approaches they are not undisputed [16]. The following are a few of the papers presenting alternatives, amendments and discussion: [25, 104, 2, 102, 100, 16, 52, 48, 1, 15, 45, 108, 80, 5, 109, 103, 23].

An important distinction regarding metric analysis is between system level and example level. On one hand, a metric can be used to rate summarizers based on many summaries that each one created. On the other hand, ratings for individual summaries are produced. Both are valid use cases. Example level correlation is performed directly using the system and human ratings for each summary. If there is more than one human rating per summary, they are aggregated first. On system level, the metric scores and the human scores are aggregated for each system, then correlation is calculated. Example level is the harder task because, at the system level, ratings of many summaries are aggregated, which reduces the impact of single errors. [56, 107, 23, 2, 16]

Metric selection in the context of Reinforcement Learning training should be performed on example level. In each step of the training the agent will explore different summaries and the policy is then updated to perform well rewarded actions more often. There is just one system, the learning agent, which needs to know how good each try was.

2.3. Evolutionary Algorithm

As motivated in section 6.4, we are using an Evolutionary Algorithm (EA) to train our ensemble. Similar to RL, EA searches for a locally optimal agent, with optimality being defined by a so-called fitness function. In EA, agents are not updated individually using gradients. Instead, multiple agents, forming a population, are compared, and the best ones are selected to create new contestants. Hence the fitness function does not reward a single stream of actions, but rather assess performance of an agent, based on any number of actions. New agents are created in every step based on mutation, which means, in the case of neural network based agents, random weight changes. We stick to this basic update mechanisms, even though other alternatives are common additions. As the best agents remain within the population, performance can stagnate at worst. We implemented our naive approach, based on common knowledge, from scratch. [61, 110, 111]

2.4. The overfitting problem

Overfitting is a well-known problem in Supervised Learning but is less prevalent in Reinforcement Learning. Well known RL applications like chess or Atari games come with perfect rewards, either winning or gathering as much points as possible. The agent cannot overfit there; 'better' simply means an improvement.

Reward exploitation is a well known issue but is usually linked to reward shaping, the process of creating intermediate rewards to ease the learning. When the intermediate reward is gamed the agent might be misguided and fail to learn good behaviour with respect to the original reward. Training is tempered, but evaluation and comparison of agents using the original reward is intact. In the field of RL-based summarization reward shaping seems not to be common.

More fundamentally, overfitting in this context describes exploitation of an end reward, which is not perfectly aligned to human desires. An automatic reward might not align equally well with human judgment through out the whole space of possible summaries. Optimization could guide the summarizer into this realms, produce good reward results, but not suit human expectations.

At the moment, only human evaluation can detect this overfitting. As discussed in section 1, the present metrics are insufficient. Moreover, if there were a mechanism to detect such behaviour it would most probably already be part of the reward.

For example overfitting has been observed during RL training with ROUGE-based rewards. [42, 35, 59, 112, 45, 33, 18, 15]

3. RL approaches

As elucidated in section 1, Reinforcement Learning has some advantages over Supervised Learning, as it operates on summary level, instead of word level, and can handle not directly differential input like human feedback or metric scores [35, 37]. Furthermore, complex summarizer structures, involving non-differential intermediate steps based on discrete words, are possible.

However, RL comes with its own downsides. Training is typically slower and less stable. Therefore it is usually combined with supervised approaches to be feasible. Pretrained Large Language Models (LLMs) or summarizers are used as the basis for RL fine-tuning. [93, 35, 33, 37]

The evaluation strategy for Reinforcement Learning approaches has to be designed with care. It should be somewhat complementary to the training approach. Reusing metrics from training is not reliable, due to the overfitting problem. Human evaluation might be the gold standard but flaws or biases in the task composition might be shared between training and evaluation, obfuscating issues.

Agent architecture (commonly a selected type of neural network), the training algorithm (for an example see PPO [113]), the installation of the task (possible actions, available information etc.), which is usually called the environment, hyper parameters (like model size, learning coefficient etc.) and reward compose a Reinforcement Learning setup. While each component holds significance, in the context of summarization, the reward is currently considered the most influential factor [38]. Exceptions exist ([38]), but the papers we reviewed mostly centered around the unshaped reward. Other components were typically selected based on RL state-of-the-art, and no issues were reported.

Therefore, we present RL approaches from the perspective of different takes on the reward.

There are two approaches directly creating a reward based on human data: RLfHF (3.1) and GAN (3.2). They differ from the metric based approach as they retrain their reward iteratively based on their generated outputs. Metric-based rewards (3.4), on the other hand, are not altered during the training process. Metrics can be used in isolation or combination to form a reward. The Two-Stage approach (3.3) has a non-differential intermediate step and treats the later stage as part of the reward. Furthermore we want to highlight that divergence from a root model and combination of RL and SL are possible methods to ground the learning (3.5).

3.1. Reinforcement Learning from Human Feedback

The main advantage lies in the alignment with human goals. However, the issue of immense amounts of human labour being required has to be solved. While theoretically every generated

summary could undergo human assessment, this becomes unfeasible due to the sheer volume of required samples. Instead a proxy is created. A reward model is trained using human feedback on a set of samples and is subsequently employed to interpolate and extrapolate ratings to the generated summaries during agent training. In that sense the reward model is a classifier- and training-based metric. The difference to the metric based approach is that the reward is not static. First, the RL-agent is trained against the current reward model. Second, new samples, which might be results of real improvement or reward overfitting, are used to update the reward model. Then the process starts over again. This prevents misalignment, as holes in the reward-model are discovered by the agent and closed by updating the reward. Due to updates of the reward model, it aligns closer than common metrics, but it still remains a proxy for human rating and perfection is for now unreachable for practical training. Nevertheless this approach is considered very promising. [59, 114, 115, 35, 45, 116]

As this approach does not rely on the availability of any pre-built components it is very flexible. However, designing a setup for the human feedback has to be done carefully (as discussed in section 2.1.1) and effects the amount of faithfulness being taken into account.

3.2. GAN

GANs consist of a generator and a discriminator. The primary objective is to generate samples indistinguishable from human examples. This process involves a two stage loop. First, the generator is trained based on feedback provided by the discriminator. If a sample is identified as non-human, it receives a negative score; if the generator successfully deceives the discriminator, it is rewarded. In the second stage the discriminator is updated in supervised fashion. The label 'human' is assigned to a set of previously collected examples, while generated output is labeled complementary. With each iteration, as the discriminator is trained on an increasingly refined set, its performance is expected to improve. This way, each time, the generator is trained using increasingly good feedback.

However, in practice stagnation and instability can occur. An exemplary failure mode is when the generator overpowers the discriminator. If the human example set is too small distinction training cannot keep up, and the training signal for the generator gets corrupted.

In the image domain, the generator is trained in a supervised manner. However, summarizing generators produce discrete words, which prevents standard back-propagation. Reinforcement Learning is used to circumvent this problem. The discriminator takes the role of the reward, which is continuously update, similar to the RLfHF approach. Instead of human judgments GANs rely on a set of human examples. As this algorithm produces outputs similar to the provided examples, desired qualities have to be well represented. Therefore, whether faithfulness can be achieved depends on the quality of the selected examples and the learning capabilities of generator and discriminator. [93, 35, 117, 118]

3.3. Two-Stage

An other approach is to prepend a new stage A to an existing module B, which takes text as input. It is equally suited, when B was trained based on a supervised loss or a reward function. A is used to pre-process any input into an intermediate text, which is then fed into B. Reward for A is assigned based on the performance change of B. The original module and its training or validation setup has become the reward for A. This approach capitalizes on RL's independence from a differentiable loss function. Continuous backpropagation from B to A wouldn't be feasible due to the involvement of discrete words in the intermediate step.

Common motivation for this approach is to utilize the advantages of different approaches in one pipe line. One scenario is to add an extractive summarization module before an abstractive one. The extractor helps to focus and to reduce run-time complexity. The abstractor ensures good linguistic quality. [50, 34]

Based on the design the additional pre-step can have positive influence on faithfulness. For example could an extraction step strip superfluous text parts, increase focus in the second stage and so reduce mix-up of distinct facts.

3.4. Metric-based

Reusing a metric as a reward or composing the reward of multiple supplementary metrics, might be the first idea, when thinking about applying RL to the summarization domain. Starting off a pretrained summarizer and closer align it to the evaluation functions.

As it is an open question, how to combine different aspect metrics to form overall quality, it is established to rely on a simple weighted linear combination. [93, 119, 120]

Since only suboptimal metrics are currently available, this approach is prone to overfitting. Therefore, it needs to be complemented with mechanisms discussed in the following section.

However this research domain has fueled metric development [33].

3.5. SL-based Grounding

A balance between supervised learning and metric-based rewards can unify both approaches, harnessing their respective advantages and mitigating the overfitting issue associated with metrics.

The first option, instead of only starting based on a pretrained summarizer, ties its agent to a root using KL-divergence [45]. This way learnt behaviour cannot diverge too much from the starting point. The KL-term is incorporated into the reward, weighted by a hyperparameter. The second approach simultaneously updates one model based on gradients from supervised loss and the reward function [39, 50, 119, 44, 42].

A more intricate combination approach is presented in [94], where either SL or RL update is performed based on a faithfulness metric's decision for each summary.

There is no downside in selecting the best available base model. Therefore it is common to see overall quality as already captured and inject special properties using metrics. Directly fine

tuning with a faithfulness metric is possible. However progress is also limited, as deviation from the root model is punished by the KL-term. The more the weight is released, the more improvement is possible, but also overfitting gets more likely. This approach might release but can probably not solve the need for more reliable metrics.

Finding a good degree of balance is the central challenge of this approach.

4. Metrics

Table 4.1.: Grouped list of metrics.

Group	Metric	Section
Baselines	Grusky	4.1.1
	Naive	4.1.2
	ROUGE	4.1.3
From AGGREFACT	DAE	4.2.1
	QuestEval	4.2.2
	SummaC	4.2.3
Addtional	DiscoScore	4.3.1
	ESTIME	4.3.2
	SGwLM	4.3.3
	BLANC	4.3.4
	Perplexity	4.3.5
	BERTScore	4.3.6
	CTC	4.3.7
ChatGPT-based	ZS, CoT, DA, Star	4.4

An overview of our selected set of metrics can be seen in table 4.1. Details will be presented in the following section, but, as a preface, it's important to emphasize that most metrics describe an approach rather than a fixed manifestation. Without altering the core concept, any Deep Learning based metric can be retrained using adjusted hyperparameters or data [29, 82]. We aimed to rely on the defaults and recommendations of the respective paper authors and will also make efforts to be clear about any deviations from this guideline. In several cases a metric suit comes including a set of slightly different approaches. We will make clear which variations we included and later also analyse their relative performance.

Furthermore, it should be noted that while a metric might be designed with a specific target quality in mind, it often correlates with multiple qualities. On one hand, these qualities are not independent; on the other hand, data and building blocks are not strictly aligned with summary qualities. Naturally, overall quality metrics should exhibit some level of correlation with other aspects.

In several cases, we needed to develop preprocessing methods for both documents and summaries to ensure flawless execution. This process will be detailed alongside each metric.

Metric selection was influenced by different aspects. First goal was to reevaluate the metric set presented in the AGGREFACT paper [47] and verify their reported results. These are presented in the section 4.2. However, we had to forgo the metric QAFactEval [81] due to practical issues regarding the setup. The included ChatGPT based metrics are presented in a separate section (4.4) alongside some discussion and proposed alterations.

To gain a better understanding of the reported results we evaluated some simpler logic-based metrics (see section 4.1) as baselines. The analysis is necessary as doubts arise regarding the dependable enhancement of elaborated metrics over baselines [16].

The additional metrics (section 4.3) were selected based on their novelty, frequent usage, and good availability.

Close to all metrics were utilized as features for the ensemble. This approach is based on the hypothesis that the ensemble Neural Network might be able to benefit from redundant information and strip away misleading signals. Investigation into the effects of feature selection, while it might be promising, is beyond the scope of this work. We excluded only the ChatGPT-based metrics and one metric version for each Perplexity and BLANC, as these would have exceeded our time frame.

In sum, the following section is supposed to present for each metric which kind it belongs to, the main ideas, relevant variations, which implementation we used and what kind of preprocessing or error handling with defaults were required. Please be aware that we do not present every algorithmic detail. Refer to the provided paper or implementation for additional information.

4.1. Baselines

In the following we present untrained logic based metrics, which rely on matching words and the length of summary and text. Their nature is deterministic. These are supposed to represent a baseline which more sophisticated metrics need to overcome significantly.

Measurements based on word-matches between original and summarized text are related to the extractivity of a summary. The relation of faithfulness to extractivity is discussed by Zhang et al. [121].

Relative and absolute summary length are associated with literal room for error and therefore might capture faithfulness. As allready discussed in section 1.1 the empty summary is trivially faithful.

4.1.1. Grusky

Three straightforward methods for assessing the relationship between text and summary are presented in 'NEWSROOM: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies' [122]. These approaches do not rely on references because they were originally designed to evaluate a collected data set rather than summarizer output.

Coverage indicates the proportion of words in a summary that also appear in the original text. Density aims to capture similarity in word order; hence, it evaluates the average length of

consecutive extracted words found in the summary. The overall length relationship, measured in words, is quantified using Compression through a simple division.

Implementations for these metrics and setup instructions are provided in the newsroom Github repository[1].

4.1.2. Naive

The metrics Length, Compression, and Length-Share are closely related to the Grusky methods presented in the last section. However, we developed these metrics before encountering Grusky's paper. Despite this, we retained them due to their minimal runtime.

We hypothesised that the Length of the summary could be correlated to faithfulness as a shorter summary leaves less room for unsupported claims. Compression follows the exact same idea as described in the last section, but implementation details might differ, as we determine word count solely relying on the amount of occurrences of consecutive white spaces. Length-Share describes the inverse of Compression.

4.1.3. ROUGE

ROUGE, proposed by Chin-Yew Lin [31], is a set of metrics designed to estimate the similarity of one text A to a suit of texts B. While originally developed to evaluate one summary against high quality references, we use it reference-less by substituting the suit B with the original text corresponding to the summary. As ROUGE is based on counting matches of words (or word groups) it has been criticised to not evaluate deeper meaning or respect synonyms. However it is still the most established metric for summary evaluation. [56, 44, 102]

We used the implementation available at huggingface[2]. ROUGE-1 and ROUGE-2 measure the unigram-overlap and the bigram-overlap. ROUGE-L, based on the longest common sub-sequence, is available in two versions, -L and -Lsum, which follow the same principle but deviate in implementation details.[3] Furthermore, we used each of these four in two modes, with and without the provided pre-processing. The Porter stemmer is used to make matches more flexible by removing word suffixes.

4.2. From AGGREFACT

This section outlines the metrics DAE, QuestEval and SummaC, which where already evaluated at the AGGREFACT benchmark [47]. All three approaches and their sub-versions are reference-free and deterministic.

[1] `https://github.com/lil-lab/newsroom`

[2] `https://huggingface.co/spaces/evaluate-metric/rouge`

[3] Further details are discussed in this Github-issue: `https://github.com/huggingface/datasets/issues/617`.

4.2.1. DAE

Dependency Arc Entailment (DAE) [12] is a trained metric based on Information Extraction. The summary is decomposed into dependency arcs, by a pretrained encoder, which are then individually examined with respect to their faithfulness. A summary is labeled unfaithful if any arc is judged unfaithful.

The corresponding implementation is accessible on GitHub[4]. Out of the available downloads, we opted for the model 'DAE_xsum_human_best_ckpt'. In their paper models trained on collected and synthetic data are analysed. This checkpoint belongs to the first category, which exhibited superior performance.

Newline characters were eliminated from both the summary and the document during our preprocessing. For ensemble training on the overall-quality data set, the additional exclusion of non-ASCII characters was necessary to prevent errors.

4.2.2. QuestEval

Scialom et al. [123] outlined a QA-based metric and provided the implementation to the public[5]. One remarkable aspect of their approach is how they integrate precision and recall simultaneously. Questions are derived from both the summary and the document, and then answered using the contrasting text. In addition, they incorporate a weighting module to assess the importance of questions. They argue that treating all questions equally could be misleading, as summaries might naturally include only a subset of the original information.

We want to point out that this metric supports both reference free and reference based evaluation. While this metric follows in principle an untrained manner, as it could solely rely on pretrained components, the authors trained everything for improved performance.

Version 0.2.4 is designed to evaluate the similarity between two texts based on their contained information. Due to a flaw in the weigther-module, version 0.1.1 is more appropriate for summarization evaluation. The newer version presents a unified score, whereas the older version provides separate scores for precision, recall, and f-score. We consider all four variations in our analysis: 0.2.4, 0.1.1-Precision, 0.1.1-Recall and 0.1.1-F-Score.

No preprocessing was applied for version 0.2.4, for 0.1.1 line-breaks were removed. Handling errors or assigning defaults was not necessary.

Tang et al. [47] selected the unweighted version 0.2.4 as the sole representative of this metric.

4.2.3. SummaC

SummaC is a NLI-based faithfulness metric and was proposed by Laban et al. [86]. Utilizing a pretrained module, for each summary sentence entailment scores to all text sentences are calculated. Per summary sentence score aggregation is performed in two variations. SummaC-ZS takes the maximum score and SummaC-Conv applies a learnt 1-D Convolution.

[4] `https://github.com/tagoyal/factuality-datasets`
[5] `https://github.com/ThomasScialom/QuestEval`

Taking the maximum score of support is plausible but is prone to outliers. The final score is calculated as average of the summary sentence scores.

In this case, one metric variation is logic-based, while the other is trained. The convolution is fine-tuned using synthetic data.

Both are available in a single GitHub repository[6]. We relied on the suggested configurations, and no additional preprocessing or error handling was necessary.

4.3. Additional

The metrics DS, ESTIME, SGwLM, BLANC, Perplexity, BERTScore and CTC, which are presented in this section, have been selected to supplement the original AGGREFACT analysis. Each of these methods is deterministic.

4.3.1. DiscoScore

While discourse coherence, well structured interdependence of sentences, is the main target of the DiscoScore (DS) metric [124], good correlation with faithfulness is also reported. We use this metric, diverging from the original intention, with the document replacing the references.

DS is an untrained logic-base metric, utilizing Information Extraction and embedding elements. Two version are proposed.

For 'Focus Difference' (Focus), tokens are assigned to so called focus transitions, which resemble the attention flow of a human reader. Each focus transition is composed of multiple tokens and each token can appear in multiple transitions. Corresponding token embeddings, derived from a pretrained module, are summed up to form focus level embeddings. The final score is calculated by aggregating the distances between the embeddings of common foci of both texts.

'Sentence Graph' (Sent) operates on sentence level instead of focus level. Pretrained sentence embeddings are updated by incorporating information from the other sentence embeddings based on the amount of shared focus transitions. These more contextualized embeddings are subsequently aggregated into a text embedding. Two texts are then assessed based on the cosine similarity of their embeddings.

DS-Focus and -Sent are both availailable int two versions: Noun (NN) and semantic entity (Entity). These are different ways of estimating focus transitions.

A Python package along with instructions is provided through a GitHub repository[7]. However, due to a limited capacity for input lengths (restricted to 512 tokens), larger samples were disregarded during Zhao's experiments. To incorporate this caveat into our results, we did deviate from this approach and created a fork[8], which adds truncation capabilities to the authors code. Following the author's suggestions, we also integrated support for Longformer [125] using a model from Huggingface. A couple of excessively long samples still required

[6] https://github.com/tingofurro/summac

[7] https://github.com/AIPHES/DiscoScore

[8] https://github.com/UntotaufUrlaub/DiscoScore_WithHandlingLongInputs

truncation though. For the truncation based versions we downloaded and incorporated the recommended models.

Additionally a negated version for each of the described approaches is added. This approach was discovered by chance and shows surprising improvement for one case, as later discussed in the analysis (section 7.2.6).

In total we evaluated 16 versions of DS: All possible combinations of Sent or Focus, NN or Entity, Truncation or Longformer and positive or negated.

We transformed all inputs to lower case letters as the sole preprocessing step. No additional error handling, resorting to defaults, was necessary.

4.3.2. ESTIME

ESTIME [107] was designed as a reference-free metric for faithfulness but has also shown strong performance in other quality aspects.

It is an untrained metric based on pre-trained embeddings. In the initial phase, embeddings are derived for all summary tokens that also appear in the text. Each token's nearest match among the original text's token embeddings is identified. If this match does not correspond to the same token, a potential inconsistency is inferred. The count of inconsistencies constitutes the final score.

Extension to this approach was later proposed by the authors to allow assessing of texts with very distinct wording [126]. It considers all summary tokens and, instead of counting matches with different tokens, the embedding distance of each match is summed up.

We considered four variations of ESTIME[9] provided by the BLANC package, version 0.3.2, dated July 25, 2023: 'alarms' with both 'include_all_tokens' set to True or False, 'soft' and 'coherence'. Preprocessing was not applied, but we addressed a few uncommon cases using a default value of zero.

4.3.3. Shannon Game with Language Model (SGwLM)

The SGwLM metric suite proposed by the paper 'Play the Shannon Game With Language Models: A Human-Free Approach to Summary Evaluation' [32] measures the amount of captured relevant information in a summary. This untrained metric utilizes token probability scores of a pretrained Large Language Model. The generation likelihood of original text in isolation and original text given the summary as prompt is compared. If the summary effectively captures the content, predicting the document becomes simpler, as minimal new information needs to be introduced. This relation is expressed in the 'Information Difference Score'. Normalizing this, utilizing the probability of predicting the document, given the document, as a lower bound, gives the 'Shannon Score'. Calculations are performed at the level of information added by a token, which is defined as the negative-log-likelihood. 'BLANCShannon' is defined as the difference in token accuracy between generating in isolation and generating given the summary.

[9] https://github.com/PrimerAI/blanc/tree/dc2417ef9305a900dfaa9d8f5dd68a0f118b524b/estime

We utilized the implementation provided as part of the BLANC package[10]. In general we did not perform preprocessing but some cases required the removal of line breaks to prevent errors. Some remaining errors had to be circumvented by relying on a default, which we chose to be zero.

This metric is related to faithfulness as any hallucinated information in the summary will probably not help generating the text rather even misguide the generator. High correlation for faithfulness and other aspects was reported.

4.3.4. BLANC

BLANC [68] is a reference free metric based on likelihood. It does not rely on training as it builds a logic upon a pretrained Large Language Model. The reconstruction performance on the original text is assessed in presence and absence of a summary. The hypothesis is that a good summary is helpful in predicting the right token for any masked token in the original text. Correlation with faithfulness and other aspects is hypothesised, as only overall good summaries should cause a beneficial effect.

There are two versions of supporting the LLM with reconstruction of the document. Blanc-Help adds the summary as prefix to the document sentence which is currently processed. Blanc-Tune finetunes the LLM for a given summary and then processes the whole text. Naturally the second approach is more resource heavy as it involves training for every assessed summary.

The corresponding reference implementation is available at Github[11]. No additional preprocessing, error handling or defaults were required.

4.3.5. Perplexity

Based on a proposal in section 2.2.1 of 'Answers Unite! Unsupervised Metrics for Reinforced Summarization Models' by Scialom et al. [44], we also used the perplexity of BERT [127] as evaluation metric. This is a reference free approach based on token likelihoods without the need for references.

Our implementation is grounded in the 'bert-base-uncased' model by Huggingface. The summary is concatenated to the text, followed by an iterative process over the summary tokens. In each subsequent step, Bert is leveraged to process the masked token and its left-hand context to compute token likelihoods. The corresponding value of the original token is then recorded. In an extended version, referred to as 'full context', right-hand tokens are also taken into account if the upper limit for BERT input has not been fully utilized. The final score is determined by the average negative log likelihood of the summary tokens.

A lower score indicates a higher likelihood of the two texts appearing together. This leads us to the hypothesis that better summaries could potentially receive lower scores. Furthermore, we explored the impact of interchanging text and summary, operating under the assumption that a text could be more readily derived if the summary imparts useful information.

[10] `https://github.com/PrimerAI/blanc`
[11] See footnote 10

Furthermore we also measured the perplexity of the summary in isolation, motivated by the assumption that internal properties like linguistic quality might be similarly sufficient to judge current summaries [49].

Empty summaries were treated with a default of zero.

4.3.6. BERTScore

A very popular [109] similarity metric is BERTScore [83]. In the context of summarization it is usually used to evaluate a summary against a set of references. However, we use the original text as stand-in for the reference. This metric is untrained and relies on embeddings of any BERT-based Large Language Model. First both texts are converted to token embeddings, then for each token the closest match from the other text is derived using cosine similarity. These best similarities are than aggregated to get the overall score. As these matches do not rely on static words, but rather capture meaning and context, the approach should be superior to word based metrics like ROUGE.

The library[12], version 0.3.13, offers precision, recall, and F-score calculation. While we have currently utilized the default model, it's important to note that there exist several robust alternatives. Consequently, in-depth analysis and comparisons should be conducted in the future.

We replaced any group of white spaces in the inputs with a blank space. We did not encounter any errors and therefore could avoid defaults.

We assume similarity between summary and text could indicate faithfulness and therefore analyse the potential of this setup.

4.3.7. CTC

The CTC metric suit [62] is developed to handle multiple scenarios and aspects utilizing estimation of information alignment. Faithfulness describes the condition that all information of a summary is aligned to the original text. The Relevance of a summary is established when all (essential) information from the source document aligns with it. Three methods have been proposed to estimate this alignments. However we only used the recommended method D, which trains a model to predict the alignment probability of each token using synthetic data. The average forms the final score.

The faithfulness version is designed without the need for references. For relevance, references are used to detect important parts in the text. However, in our case, we substituted the references with the document.

We utilized the promoted implementation[13], available on Github, in version 0.1.3. For the Faithfulness score two models were available: XSUM and CNNDM. Only one model, CNNDM, is provided for relevance.

[12] `https://github.com/Tiiiger/bert_score`

[13] `https://github.com/tanyuqian/ctc-gen-eval`

In the preprocessing stage, we replaced each group of white spaces with a single blank. While working with the ensemble training dataset, we addressed errors that arose in extremely rare cases by applying a default value of zero.

4.4. ChatGPT-based Metrics

Our investigation considers four recent ChatGPT-based metrics which were also benchmarked in the AGGREFACT paper. These are built on top of the API provided by OPENAI, which is by default not deterministic. This is an odd property for a summary evaluation metric, but we did not deviate from the default settings, as this was not proposed in the original papers.

The metrics ZS and CoT were proposed by Luo et al. [128] and perform a binary faithfulness decision. DA and Star were introduced by Wang et al. [129] and assess the degree of faithfulness using a wider range of possible scores. Each metric consist of a prompt, which is supplemented with article and summary, and a heuristic to turn the free text response into a score. The exact prompt templates are documented in Appendix A.1.

The ZS metric is quite straight forward. A binary decision is requested and response options (yes and no) for faithful and unfaithful are provided. CoT is more complex. Following the assumption that reasoning might improve the answer, ChatGPT is prompted to provide some explanations and then a binary score in its response. The final decision pattern (yes and no) is again included.

DA prompts ChatGPT to assess the faithfulness on a range from zero to 100. Star request judgment on a scale from one to five. Higher means better in both cases.

These approaches do not incorporate the multi-turn capabilities of ChatGPT.

These metrics do not include references into their prompt patterns, however this would be possible [129]. These are untrained metrics, as only logic is utilized to process the answer of the pretrained ChatGPT-component.

Truncation of some rare long texts was necessary to respect the input constraint of the used 'gpt-3.5-turbo' model.

The following subsections will present the utilized heuristics, discuss their performance and promote two possible adjustments: Sum-Step and Retries.

4.4.1. Heuristics

Neither the original papers nor the AGGREFACT paper included their used heuristics. As of early August 2023, the AGGREFACT repository does also not yet contain them, although they are planned to be included eventually.[14] Heuristics for DA and Star are available in the corresponding GitHub project.[15] Due to a lack of available information, we developed our own heuristic for the metrics ZS and CoT.

[14] https://github.com/Liyan06/AggreFact/issues/6

[15] https://github.com/krystalan/chatgpt_as_nlg_evaluator/issues/3

In the case of ZS, we assess whether the response is 'yes' or 'no' while ignoring punctuation, letter casing and quotation marks. We have retained this methodology for CoT; however, given the anticipated expanded nature of the responses due to the reasoning requirements, we additionally addressed instances of 'yes' and 'no' at the beginning and end of verbose answers. Additionally, we incorporated the presence of the key terms 'consistent' and 'inconsistent,' which are synonymous with 'faithful' and 'unfaithful' and were used in the prompt.

The proposed heuristic for DA returns the first occurring number in the response. The proposed approach for Star checks whether the first word is a number between one and five, either as word or digit.

All these approaches rely on returning a default value if they could not successfully extract an assignment or score. We investigate the influence of default selection in our analysis by comparing two versions for each approach, one resorting to faithful and the other to unfaithful.

4.4.2. Sum-Step

As we suspect the simple heuristics, especially for CoT, to be insufficient, we propose to enhance them, making use of ChatGPT again. We propose to take a second turn in the conversation and send a prompt to summarize the given answer into a fixed format. The hypothesis is that this way the default rate can be reduced and therefor the performance improved, without introducing any new module or complexity. We reused the previously proposed logics, including the defaults, to process these second responses.

We implemented and analyzed this new approach for every metric except ZS, as it was not dependent on the default in many cases regardless.

In the following the precise additional prompts are listed:

CoT

Please sum up your answer with either "Yes." for consistent or "No." for inconsistent.

DA

Please repeat only(!) the score number.

Star

We designed two prompts, a general one and the other to specifically target the problem of a decimal point number of stars. By default we did use the first option, while we selected the second if a regex detected the presence of a decimal point number in the original response.

- General:

  Please repeat only(!) the star score, following the pattern "x stars" where x is the number of stars you assigned.

- Decimal point:

  Please repeat only(!) the star score, following the pattern "x stars" where x is the integer number of stars you assigned. If you assigned a float number of stars, please make a decision.

4.4.3. Retries

We propose a further approach of improvement, which we also include into the later analysis.

Due to the probabilistic nature of the ChatGPT-API the required response pattern might be broken by bad luck. Therefore we propose, instead of defaulting, repeating the request again to provoke a better suited answer. This approach is of low complexity as no prompt has to be designed and only little logic needs to be added. We decided to attempt five retries before resorting to the default value.

5. Benchmark: AGGREFACT

AGGREFACT [47] is a benchmark designed to assess the reliability of reference free metrics in evaluating the faithfulness of a summary to a source document. It consolidates nine established [130] datasets into a unified framework. Each component includes human judgments on the quality of model generated summaries. The selected systems are mainly transformer-based and range from SOTA to older architectures. Various error types have been aligned, conflicting labels have been corrected, duplicates have been removed, and diverse labeling schemes have been converted to binary format. The proposed evaluation is performed using balanced accuracy to respect the differing data share of positive and negative label.

The texts mainly represent the news domain, sourced from the established [5] CNN/DM [131] and XSUM [132] data sets. Tang et al. derived separate score for each of these datasets, as they might represent different properties.

Summaries within the first dataset tend to be extractive, while those in the second dataset lean towards being more abstractive [47]. Moreover, each dataset displays a unique distribution of faithfulness error types [47, 12]. These datasets were constructed using different approaches to reference extraction [20], leading to variations in gold label styles that likely impacted model training. Consequently, even the summaries generated by models exhibit distinct characteristics. Furthermore, the two datasets differ in terms of length ratios between summaries and source documents. As illustrated in Figure 5.1, the CNN/DM dataset spans a wide spectrum from highly compressed to more lenient summaries, whereas XSUM contains fewer instances of extreme samples.

The AGGREFACT dataset serves as the foundation for our evaluations due to its broad coverage of data sets, systems, and summaries. Nevertheless, it's recognized that its size remains relatively small, comprising around ten thousand document-summary pairs.

Another comprehensive option is the TRUE benchmark [23], which we neglected as it encompasses multiple tasks and is thus less tailored to summarization. It also employs the binary annotation format as a unification approach, bolstering confidence in that design choice. The previously widely-used DUC/TAC datasets are considered unrepresentative due to their age [104, 2].

Tang et al. [47] propose multiple evaluations based on the AGGREFACT data. We do not consider their analysis regarding different faithfulness errors and declining metric performances for outputs of more recent systems. We rely on the overall performance analysis provided in the corresponding Github repository[1].

[1] `https://github.com/Liyan06/AggreFact`

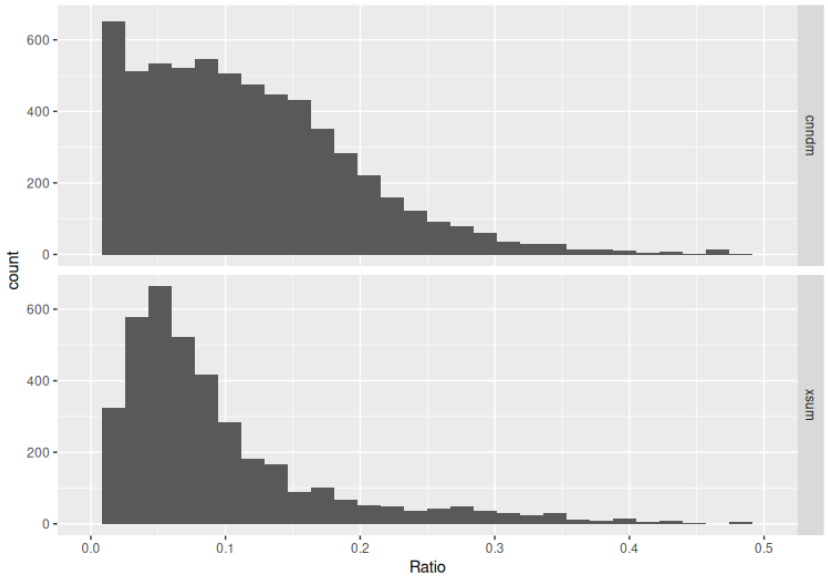

Figure 5.1.: Distribution of length ratios between summary and document for the CNN/DM and XSUM parts of AGGREFACT

Not all metrics utilize a binary decision format for faithfulness judgment, wider scoring ranges are also common. The authors propose to convert these using a threshold as decision boundary, which is tuned using an included validation set. Threshold tuning is available individually for different subgroups of the benchmark. However, the overall evaluation performs tuning only on the SOTA subset, to not overly abuse older summaries, which were deemed easier.

Due the utilized binary format and balanced accuracy, the performance of random classifier is expected to be close to 0.5 [47]. The performance distribution of a 100 runs of a random functions, depicted in Figure 5.2, supports this expectation but also highlights the potential deviation.

In our analysis (section 7) we will repeat some evaluations done by Tang et al. to verify their results. Section 4.2 provides more information about the selected metrics. Additionally, we will add results for some baselines (introduced in section 4.1) to allow the assessment of the explanatory power. Furthermore we will include some additional metrics (introduced in section 4.3) to give RL practitioners information about a broader range of reward building blocks. Influence of different heuristics onto the performance of ChatGPT-based metrics is analysed in section 7.2.12.

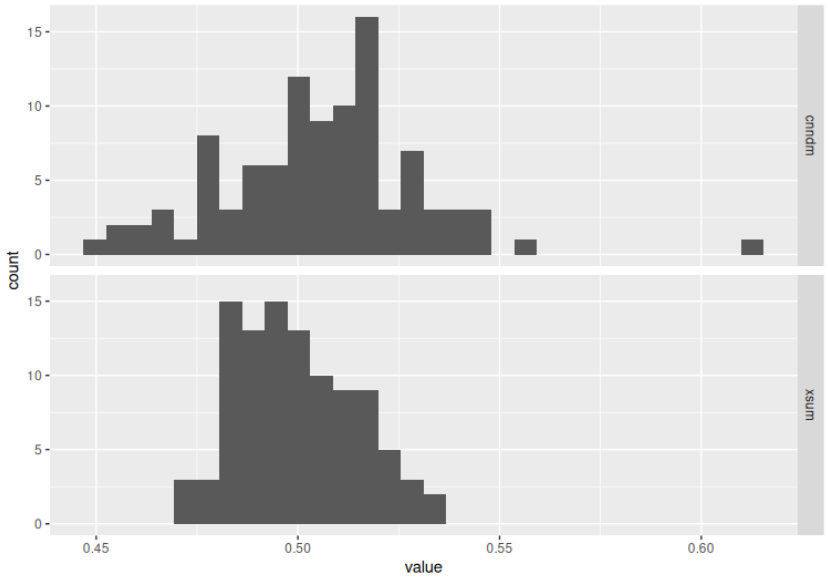

Figure 5.2.: Performance (balanced accuracy) for a 100 runs of a random function on AGGREFACT

6. Ensemble

As discussed in section 1.1 building a reward solely focused on faithfulness can hardly be beneficial as faithfulness alone has a trivial solution. Therefore a reward needs to balance faithfulness with other qualities. However, how such a combination should look like is unclear. The relationships among these qualities are not clear. Some might add up, others might be opposing each other, proposing the need for compromises [56, 85, 19, 103, 48].

Therefore we argue RL-training for overall summary quality is the more reliable approach and should also improve faithfulness eventually. Other approaches bare the risk that high faithfulness scores might have been achieved by totally sacrificing other qualities.

It is important to emphasize that an equivalent level of overall quality can be attained through significantly different performance in terms of individual qualities. Since the relationships among these aspects and their contribution to overall quality are not yet fully understood, similar levels could be reached using diverse combinations of individual strengths. Moreover, detecting such shifts in quality during final evaluations might prove challenging due to the limitations of both human judgment and metric-based assessment. Therefore, a thorough evaluation is essential.

We argue a RL approach should train for overall quality and evaluate for faithfulness. If special focus on faithfulness is required, a shift could be applied once the best level of overall quality is achieved. Other aspects should be monitored carefully.

However, capturing overall quality is still an open challenge for current metrics [56]. While it is clear that not only metrics can be used to compose a reward, as seen in section 3, we decided to focus on metric improvement, because we think a metric, which respects overall quality and faithfulness, is a powerful building block and evaluation tool.

The described need for combining and balancing diverse building blocks into a unified reward can be frequently observed in Reinforcement Learning papers [42, 57, 44, 99, 40, 133]. Furthermore many papers ([81, 23, 47, 49, 103, 48, 18]) see great potential in learning a combination of different aspects. Also different metrics, as they disagree even regarding the same property, might be combined for increased robustness [47, 52, 29, 109, 98, 81, 23].

Hence, we combine the presented metrics (section 4) by means of a neural learner optimised for overall summary quality. As faithfulness is part of overall quality and faithfulness metrics are included, we hope to improve faithfulness alongside. Success of our training regarding overall quality will be discussed in section 6.5. Whether faithfulness could also be improved will be assessed in the analysis chapter (7). The training data is presented in section 6.1. The

architecture for the ensemble network and a special initialisation approach are outlined in section 6.3. Section 6.4 covers the details regarding the Evolutionary training (2.3) process.

6.1. Data

As we are not aware of any established datasets of human judgments for news summaries, we selected the data sets of Goyal et al. [30] and Stiennon et al. [45]. By combining two datasets we hope to cover a broad range of text and summary styles.

The dataset outlined in [30] "News Summarization and Evaluation in the Era of GPT-3" comprises news articles collected from CNN and BBC in 2022. The authors collected around a thousand model-generated summaries, which were evaluated by humans using the Best-Worst-Rating (2.1.1) method. This dataset is expected to be of high quality due to its recent acquisition and the authors' efforts to mitigate bias.

As no data splits for training, validation, and test were provided, we created our own, with the shares 0.6, 0.2 and 0.2 respectively. Samples were assigned pseudo randomly. Hence, there should be no bias and the splits can be recreated. We used a hash value of the article to provide a pseudo random decision. However we did not end up using the test split for now.

The second dataset was composed by Stiennon et al. [45]. It composes texts from reddit.com and CNN, therefore it aligns less precisely with our target domain. Nevertheless, we considered it valuable due to its potential to provide scale. A portion of the data was rated using an absolute scale, while the rest was evaluated using comparisons. Pre-defined data splits were used; the test split has not been utilized at this point.

In sum did we utilize three portions of data: Data-Goyal, Data-Stiennon-Compares and Data-Stiennon-Scale.

6.2. Method of Performance Assessment

The correlation method was chosen to evaluate performance on the overall quality dataset, given that this dataset employs fine grained labels and correlation has been well-established for such analyses [26]. Consequently, we transformed Data-Goyal and Data-Stiennon-Compares from relative ratings to scales using ELO, as explained in section 2.1.1.

The correlation scores of all three datasets are aggregated through averaging. Additionally, the scores of individual sub-graphs within each dataset are also averaged. This treatment assigns equal importance to the three datasets. While averaging correlations might not be the ideal mathematical approach, it aligns with common practice in summarization research [106].

Furthermore, comparison between various ensembles and baselines was conducted using the absolute value of correlation. This approach is employed because negating metric scores is straightforward, making a high negative correlation equally valuable as a positive one.

An implementation caveat is that correlation values for some sub-graphs had to be replaced with a default of zero. During the training process the ensemble could learn to assign every sample the same score. In that case correlation would not be defined. We assigned a default of zero, the worst possible value, to enforce learning of decisions.

6.3. Ensemble architecture

Due to the fact that the relation between different quality aspects and divers metrics might be rather complex a Neural Network is selected as learning backbone. As no raw input, but only a couple of metric values, have to be processed a simple three layer fully connected setup is utilized. The intermediate layers have sizes of 20 and 10. ReLU is employed as the activation function for all layers. The output of the final layer neuron remains unscaled, as correlation is independent of scale.

We considered feature normalisation as potentially useful. Different metrics operate on different scales, especially the naive metrics like Length are rather untypical. Learning to balance these might take additional time. Bringing these to similar ranges might be beneficial for ease of training.

Two initialisation approaches were considered. The first one is standard PyTorch initialisation. The other explicitly sets weights such that a network ignores all but one feature. At the beginning its predictions closely resemble the input feature. The concept aims to improve prediction by progressively incorporating information from additional features, commencing with a base feature. This might provide a smooth start for ensemble training and establish a base level of performance. However it could also trap candidates in unfortunate local optima.

The influence of the intialisation approach and utilization of feature normalisation will be analysed in section 6.5.1.

6.4. Training Setup

As we chose correlation as the performance measure, we opted to utilize the Evolutionary Algorithm for training. Supervised Learning is not suitable since correlation cannot be differentiated. Although Reinforcement Learning (RL) could be a potential approach, it was disregarded due to the metric score's single-step calculation, which opposes RL's focus on the allocation of rewards across multiple steps.

One special feature was added to the implementation of the EA: Mutation Escalation. Based on the age of an ensemble within the population, the mutation range is altered. The assumption is that a model persisting in the population for an extended duration could occupy a position in the weight space, serving as a rough starting point for further improvement. Otherwise, it would have been displaced by its offspring. Therefore we randomly decide to apply one of two changes. The first assumes a local optima and increase the range of mutation to 'jump' out of it. The second assumes that a smaller and therefore better covered search space might be beneficial to detect an improvement.

Not all metric versions could be included as feature for the ensemble due to limited amounts of time and computation power. The ChatGPT-based methods were excluded as they are throttled by the API rate limit. BLANC-Tune was omitted due to its significantly slower processing compared to BLANC-Help. The authors assert a strong correlation between both methods in the repository description. We deemed it unnecessary to include all versions of perplexity. We excluded the version that swaps text and summary based on full context due to its slower performance.

Additionally, we conducted training and validation on a subset of the available data. The samples in Data-Stiennon-Scale were pseudo-randomly selected. For Data-Goyal and Data-Stiennon-Compares, we selected sub-graphs instead of individual samples to preserve the integrity of ELO and correlation calculations. The resulting training and validation set are each roughly about 10k samples big. Table 6.1 shows the share for each dataset.

Table 6.1.: Data share for Training and Validation on the three overall quality datasets

	Training	Validation
Data-Goyal	100%	100%
Data-Stiennon-Compares	10%	50%
Data-Stiennon-Scale	50%	100%

Given that the Evolutionary Algorithm approach involves multiple agents, exploring diverse architectures within a single training session poses no issues. The ensemble comprises 48 metrics as features. For each metric, we introduced two manually initialized models: one with feature normalization and one without. Additionally, we incorporated 104 randomly initialized models, which were randomly assigned to either utilize or disregard feature normalization. In total, this resulted in 200 models.

Seven training runs were executed in parallel to mitigate the large influence of random components. Each run was scheduled to progress for 100 update steps and then prolonged for another 75. In each step the bottom half of the population was discarded. A new mutated model was added for each remaining one. Mutation power was set to 0.04 and mutation-escalation to 0.004.

6.5. Training Progress

Figure 6.1 depicts the development during each training run. The curve depicts the performance of the best model for every step. A successful ensemble has to be better than its individual parts. Therefore we marked the performance of the best feature in each plot. These are not exactly the same because data processing using ELO involves some randomness, but they are roughly at the same level. Every run shows some learning with respect to the training data set, as the initial performance is improved and the baseline is outdone. The initial step already performs quite well due to the special initialisation.

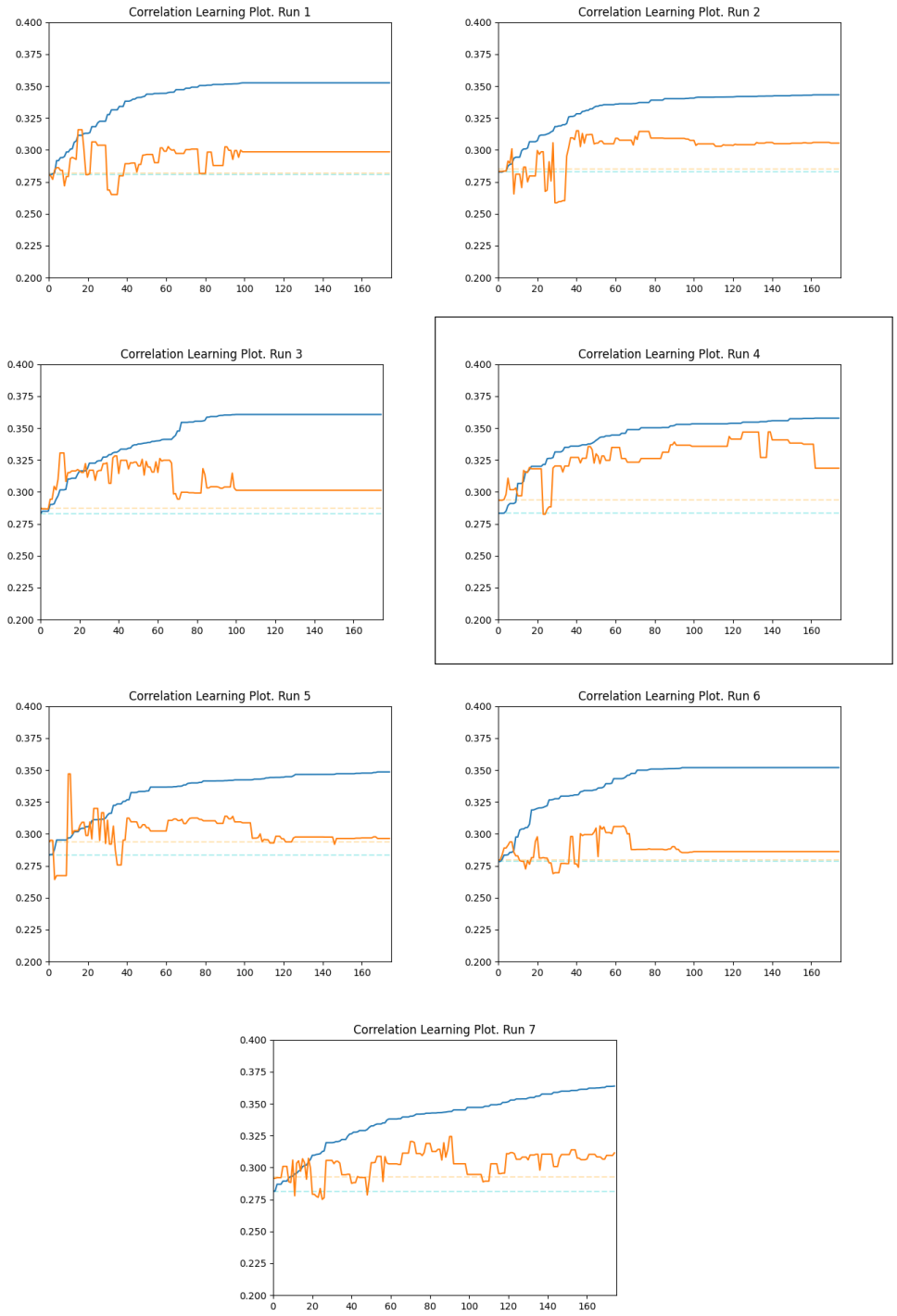

Figure 6.1.: Ensemble Training progress for each run. Blue line: On the training set. Orange line: On the validation set. Orange dashed line: Best feature on the training set. Blue dashed line: Best feature on the validation set. Run 4 is highlighted as it was selected for the final ensemble.

The best performance value and model, with respect to validation, is recorded during each run. The corresponding values are listed in table 6.2. The best of these models, which belongs to run four, is selected as the final ensemble model. Progress on the training set translates to some degree to the validation set, where we also record some gains. However, the plot for run four shows that progress on the validation set might already be stagnating.

Further analysing the validation performance of all runs shows that translation of training gains to validation is quite brittle. This might be an indicator that the training data set is to small.

In conclusion, we consider the training to have been successful; however, we emphasize potential issues related to the size of the dataset.

Table 6.2.: Ensemble performance on the validation set. Start vs best model. Run 4 is highlighted as it was selected for the final ensemble.

Run	start	best
1	-280	-315
2	-283	-315
3	-287	-330
4	**-293**	**-347**
5	-293	346
6	-279	-306
7	-291	-324

6.5.1. Analysis of Feature-Normalisation and Initialisation

Table 6.3.: Architecture features used by the best model in each run. True means the approach is used. False means the approach is not used. Run 4 is highlighted as it was selected for the final ensemble.

Run	feature-based initialisation	feature normalisation
1	True	True
2	True	True
3	True	True
4	**False**	**True**
5	True	True
6	True	True
7	True	True

Table 6.3 shows which architecture the best model, with respect to training, in each run is based on. Hence, input normalisation seems to be overall useful. The special initialisation

is also quite successive, however, the run, which provides the best model, with respect to validation, shows strong performance for random initialised models. Therefore it remains unclear which is the better approach for now.

7. Analysis

In the following the results of our experiments are presented and analysed. Section 7.1 analysis whether our set up did reproduce the results of Tang et al. and shows that the overall trends are very close. However our result for ChatGPT-CoT is dramatically better than originally reported. The diverse versions per metric are analysed in section 7.2. Per subsection a table presents the balanced accuracies and will then be analysis. Section 7.3 compares different metrics against each other based on the best version per metric and data split. A surprising high performance of the baseline metrics, which is hardly surpassed by any metric, is reported. Furthermore disappointing performance of our ensemble is observed. Motivated by weak ensemble performance we compare performances of metrics on the Training dataset to AGGREFACT in section 7.4 and see significant differences between the CNN/DM and XSUM split.

7.1. Tang vs Us

Table 7.1.: This table contrasts balanced accuracy scores reported by Tang et al. to the results of our experiments. We report values for ChatGPT metrics using the described corresponding heuristic and the default value for unfaithful (*). We do not know which heuristics were used by Tang et al. The values are taken as reported in their paper. The value is highlighted if our score differs by more than 0.01.

Metric	subversion	CNN/DM		XSUM	
		Tang	our	Tang	our
DAE [1]		0.654	**0.630**	0.702	**0.667**
QuestEval	0.2.4	0.702	0.702	0.595	0.594
SummaC	ZS	0.640	0.640	0.564	**0.539**
SummaC	Conv	0.610	0.610	0.650	**0.635**
ChatGPT-ZS	*	0.563	0.562	0.627	**0.657**
ChatGPT-CoT	*	0.525	0.532	0.559	**0.571**
ChatGPT-DA	*	0.537	0.544	0.549	0.546
ChatGPT-Star	*	0.563	**0.535**	0.578	**0.622**

[1] Values for DAE on XSUM are probably an overestimation of general capabilities as DAE was trained on XSUM based data.

Table 7.2.: This table contrasts balanced accuracy scores reported by Tang et al. to the results of our enhanced ChatGPT-based metrics. ChatGPT-based metrics use our best heuristics (**). We do not know which heuristics were used by Tang et al. The values are taken as reported in their paper. The value is highlighted if our score differs by more than 0.01 from our score using the basic heuristic.

Metric	subversion	CNN/DM		XSUM	
		Tang	our	Tang	our
ChatGPT-CoT	**	0.525	**0.543**	0.559	**0.661**
ChatGPT-DA	**	0.537	0.546	0.549	0.548
ChatGPT-Star	**	0.563	0.535	0.578	0.630

We report the performance of our setups of DAE, QuestEval, SummaC, ChatGPT-ZS, ChatGPT-CoT, ChatGPT-DA and ChatGPT-Star in table 7.1 alongside the results reported by Tang et al. The depictated values for the ChatGPT-based metrics are derived using the basic heuristic.

Our DAE setup exhibits slightly inferior performance, which we attribute to a suboptimal preprocessing pipeline. Also SummaC might be missing an additional preprocessing step to get on equal level. Inspecting the comparison on XSUM, shows that our ChatGPT-based metrics perform slightly better. We assume the reason could be slight divergence in the heuristics. However this should be verified once the heuristics used by Tang et al. are published.

Overall trends are quite close. Therefore we conclude that neither our setup nor theirs has any hidden characteristics which would alter results.

However inspection of the results achieved using our improved heuristics (table 7.2) shows that ChatGPT-CoT can be a very potent metric for the XSUM dataset. If DAE is excluded, as its value might be an overestimation due to leakage between training and test data, it is the highest rated metric for XSUM.

Analysis, using the improved heuristic for CoT, aligns with findings stated in the original paper of Luo et al. [128]. CoT was originally reported to be superior to ZS, which was not reflected in the values reported by Tang et al. Possible discrepancies in the heuristics should be cleared once Luo et al. and Tang at al. have published their used heuristics. For a fair comparison the best available heuristic should be used.

7.2. Intra-Metric Analysis

7.2.1. ROUGE

Performance values for ROUGE are reported in table 7.3 and show that ROUGE is significantly stronger on the CNN/DM dataset than on XSUM. On the later the performance is close to or even below the expected level for a random function. This result reflects the claim that

Table 7.3.: Analysis of the proposed versions of ROUGE. Highest values with and without Stemming is highlighted for CNN/DM and XSUM respectively

version	CNN/DM	XSUM
1	0.627	0.466
2	**0.668**	**0.536**
L	0.656	0.481
Lsum	0.656	0.476
1-Stemming	0.629	0.461
2-Stemming	**0.675**	**0.535**
L-Stemming	0.671	0.462
Lsum-Stemming	0.671	0.468

ROUGE is less suited for more abstractive texts. Using the Stemming-Preprocessing does not make a big difference. ROUGE-2 with or without Stemming seems overall to be the best version.

7.2.2. QuestEval

Table 7.4.: Balanced accuracy for all versions of QuestEval for CNN/DM and XSUM. Best performance on each data split is highlighted.

version	CNN/DM	XSUM
0.2.4	**0.702**	0.594
0.1.1-Precision	0.650	**0.642**
0.1.1-Recall	0.622	0.499
0.1.1-F-Score	0.653	0.618

Table 7.4 shows that overall performance on CNN/DM is better than on XSUM.

Selecting the best version of QuestEval heavily depends on the dataset characteristics. Version 0.2.4 is the strongest on the CNN/DM dataset and version 0.1.1-Precision on XSUM. We assume the difference is caused by diverging relation of summary to text length. CNN/DM incorporates a bigger share of less compressed summaries. This characteristics fits version 0.2.4 which was proposed to be optimal for assessing texts of similar information amount. On XSUM the advantages of the weigther, tailored to summarization handling the mismatch of information amounts, makes up the disadvantage of an outdated version.

As 0.1.1-Precision also performs reasonably well on CNN/DM we deem it the recommended version when data characteristics are unclear.

Table 7.5.: Balanced accuracy for both versions of SummaC for CNN/DM and XSUM. Best performance on each data split is highlighted.

version	CNN/DM	XSUM
ZS	**0.640**	0.539
Conv	0.610	**0.635**

7.2.3. SummaC

Performance values for SummaC are reported in table 7.5. Which version to choose of SummaC heavily depends on the data characteristics as each version is the best for one dataset. Overall SummaC-Conv seems to be more stable.

7.2.4. Naive and Grusky

Table 7.6.: Balanced accuracy for the Naive metrics and metrics proposed by Grusky for CNN/DM and XSUM. Best performance of Naive and Grusky on each data split is highlighted.

metric	version	CNN/DM	XSUM
Naive	Length	0.554	0.535
Naive	Length-share	**0.634**	0.500
Naive	Compression	0.485	**0.587**
Grusky	Coverage	0.666	**0.621**
Grusky	Density	**0.680**	0.598
Grusky	Compression	0.488	0.586

Table 7.6 shows the performance of all metrics proposed by Grusky and in our Naive set.

Assessing length as sole metric for faithfulness is better than random for CNN/DM, but not very good.

The relation of length of summary and text as expressed by Naive-Length-Share, Naive-Compression and Grusky-Compression is already better. Our Naive-Compression implementation performs as good as Grusky's. Surprisingly the best methods for CNN/DM and XSUM are not the same but rather the inverse of each other. This highlights that both dataset have very different characteristics.

Overall the Grusky methods based on word matches are better than the methods based on length. Coverage is best on XSUM while Density is best on CNN/DM, again highlighting the data set differences.

Table 7.7.: Balanced accuracy for all versions of BLANC for CNN/DM and XSUM

Version	CNN/DM	XSUM
Help	0.666	0.471
Tune	0.681	0.520

7.2.5. BLANC

Based on Table 7.7, BLANC-Tune performs better than -Help. However both versions perform very poorly on the XSUM dataset.

7.2.6. DiscoScore

Table 7.8.: Balanced accuracy for all versions of DiscoScore for CNN/DM and XSUM. Values significantly above random level are highlighted.

Longformer / Truncation	Version	positive metric score		negated metric score	
		CNN/DM	XSUM	CNN/DM	XSUM
Truncation	Sent + NN	0.499	0.486	0.484	**0.574**
Truncation	Sent + Entity	0.499	0.486	0.484	**0.574**
Truncation	Focus + NN	0.507	**0.591**	0.536	0.498
Truncation	Focus + Entity	0.507	**0.591**	0.536	0.498
Longformer	Sent + NN	0.498	0.502	0.521	**0.601**
Longformer	Sent + Entity	0.498	0.502	0.521	**0.601**
Longformer	Focus + NN	0.437	**0.575**	0.516	0.500
Longformer	Focus + Entity	0.437	**0.575**	0.516	0.500

Performances for different versions of DiscoScore are reported in table 7.8 and show that the selection of NN and Entity has no influence at all. All versions perform about as good as the random function on the CNN/DM dataset.

When considering only the originally proposed versions without negated scores, Focus demonstrates certain capabilities, while Sent performs at a random level. Furthermore the Longformer version of Focus performs worse, probably because the model used in the truncated version is better suited.

However if the negated score is considered the image flips. As expected Focus-Negated is not useful. Surprisingly Sent-Negated is quite strong. It shows overall the best accuracy among all 16 proposed approaches. Currently we can not explain this gain, which might be caused by a strong negative correlation. This might be an artefact actually caused by bad performance, as DiscoScore was neither designed to be used for faithfulness nor in a reference-free manner. Additionally the Longformer version suddenly outperforms truncation.

7.2.7. BERTScore

Table 7.9.: Balanced accuracy for all versions of BERTScore for CNN/DM and XSUM

Version	CNN/DM	XSUM
Precision	0.687	0.600
Recall	0.688	0.478
F1	0.679	0.511

Experiment results for BERTScore are presented in table 7.9.

The performance on CNN/DM is better than on XSUM. On the former all versions perform about equally good. This might be a hint that there is a strong overlap of faithful summaries and overall good summaries. On XSUM only precision is above random level.

7.2.8. CTC

Table 7.10.: Balanced accuracy for all versions of CTC for CNN/DM and XSUM

Version	CNN/DM	XSUM
D-CNNDM-Consistency	0.682	0.594
D-CNNDM-Relevance	0.669	0.590
D-XSUM-Consistency	0.688	0.631

Table 7.10 shows that all versions of CTC perform above the random function. Consistency is better than Relevance, but not as much as expected. The XSUM-based version performs best on both datasets.

7.2.9. SGwLM

Table 7.11.: Balanced accuracy for all versions of SGwLM for CNN/DM and XSUM

Version	CNN/DM	XSUM
ShannonScore	0.710	0.492
InfoDiff	0.578	0.570
BlancShannon	0.678	0.484

This analysis is based on table 7.11. While the three versions show very different performance On CNN/DM, all versions are above random level. ShannonScore is the best here. On XSUM only InfoDiff, the weakest on CNN/DM, is above random. So, each data set requires its

own metric version and the elaborated approaches ShannonScore and BlancShannon provide no benefit on XSUM.

7.2.10. Perplexity

Table 7.12.: Balanced accuracy for all versions of Perplexity for CNN/DM and XSUM. We highlight values above 0.54, which we deemed sufficiently better than random level

Considered texts	context	Version	CNN/DM	XSUM
both	left	positive	0.502	0.503
both	full	positive	0.531	0.522
both	left	negated	0.519	**0.558**
both	full	negated	0.492	0.490
summary only	left	positive	0.499	0.487
summary only	full	positive	0.500	0.504
summary only	left	negated	0.498	0.470
summary only	full	negated	0.538	0.500
both, swapped	left	positive	0.421	0.517
both, swapped	full	positive	**0.552**	0.511
both, swapped	left	negated	0.510	**0.569**
both, swapped	full	negated	0.524	0.486

Overall perplexity performs quite poorly, as depictated in table 7.12. Considering only the summary provides no information regarding faithfulness at all. Negated-Left-Context provides some information for XSUM pairs, independent of swapping. Positive-Swapped-Full-Context shows some potential for CNN/DM data.

7.2.11. ESTIME

Table 7.13.: Balanced accuracy for all versions of ESTIME for CNN/DM and XSUM

Version	CNN/DM	XSUM
soft	0.690	0.591
coherence	0.606	0.523
alarms (all Tokens)	0.484	0.496
alarms (not all Tokens)	0.469	0.532

Results in table 7.13 show that both Alarm-based versions perform only at random level. Coherence only works on CNN/DM. Soft seems to be the best option by a large margin.

7.2.12. ChatGPT-based Metrics

Table 7.14.: Balanced accuracy for all heuristic versions of ChatGPT-CoT, -DA and -Star. We highlight the best value of every line, except when all values lie within 0.01. In each column value pairs which differ based on different default choice by more than 0.01 are underlined

Dataset	Default	Metric	Heuristic	Sum-Step	Retries	Sum-Step and Retries
CNN/DM	False	CoT	0.532	**0.543**	0.527	**0.543**
	False	DA	0.544	0.546	0.545	0.546
	False	Star	**0.535**	0.518	0.503	0.518
	True	CoT	0.527	**0.543**	0.527	**0.543**
	True	DA	0.545	0.546	0.545	0.546
	True	Star	0.503	**0.518**	0.503	**0.518**
XSUM	False	CoT	0.571	**0.661**	0.571	**0.661**
	False	DA	0.546	0.548	0.544	0.548
	False	Star	0.622	0.619	**0.630**	0.621
	True	CoT	0.571	**0.661**	0.571	**0.661**
	True	DA	0.546	0.548	0.544	0.548
	True	Star	0.619	0.621	**0.630**	0.621

The performance of different heursitcs for the metrics ChatGPT-Cot, -DA and Star is reported in table 7.14

The selected default has some influence on the performance but only to a minor degree.

Our proposed improvements do not provide significant gains on the CNN/DM dataset. Also DA and Star on XSUM do not gain anything from an altered heuristic. However, CoT on XSUM is significantly improved by the Sum-Step. The hypothesis is that CoT has the most complex response pattern, as it not only provides the final scoring but also some explanations, and therefore requires a more powerful heuristic. Simple answer patterns are sufficiently convertible for simple heuristics and therefore enhancement is superfluous.

Complex reasoning and response translation seems to be worth the effort as CoT performs better than DA and Star given the right heuristic.

7.2.13. Ensemble

We compare the normal and the negated ensemble in table 7.15 as we treated negative and positive correlation equally in the training process, because we think high values for both can have the same expressive power. Surprisingly not one version is good for both datasets. CNN/DM requires the negated version, while XSUM the normal one. However performance on XSUM is only slightly above random level.

Table 7.15.: Balanced accuracy for CNN/DM and XSUM of the original and the negated ensemble metric.

Version	CNN/DM	XSUM
normal	0.482	0.557
negated	0.691	0.519

7.3. Inter-Metric Analysis

Comparison between the presented metric suites is carried out based on the information depicted in table 7.16. The reported accuracy per metric corresponds to the best available version for each dataset, as determined in the previous section. Furthermore, the accuracy is also presented in the form of buckets to illustrate trends more clearly. Buckets are calculated using the formula: $bucket = \lfloor (accuracy - 0.5)/2 * 100 \rfloor$, resulting in a total of 25 buckets. Bucket 0 represents accuracy at the random level, while bucket 25 signifies a perfect accuracy of 100%.

The strongest baseline is presented by Grusky. It achieves strong performance with 68% and 62% balanced accuracy on CNN/DM and XSUM respectively. That is surprisingly high for simple methods based on word matches. Grusky surpasses ROUGE on CNN by a small margin. However ROUGE suffers from the abstractive nature of XSUM in the absence of references, while Grusky is surprisingly less affected.

Only few of the elaborated metrics surpass this baseline significantly, indicated in the table by a higher bucket. On CNN/DM it is only SGwLM and Questeval. The later is also above the baseline on XSUM, alongside of DAE and ChatGPT-ZS and -CoT. Therefore Questeval seems to be the best metric when data characteristics are not clear. The good performance of binary ChatGPT-based methods motivate more research in this direction.

The fact that elaborated metrics do not surpass the simple Grusky baseline by a large margin is quite surprising. Either elaborated metrics only manage to capture about the same word based information or different information which is however not more useful. While the second might be the case the ensemble did not manage to combine the different approaches to achieve any improved performance. Accuracy on CNN/DM is about as good as the baseline. Performance on XSUM has diminished.

Another interpretation for the close performance of Grusky and the best metrics is that the AGGREFACT dataset might not cover enough different summary style to distinguish simple approaches from elaborated ones.

Both options are worrisome: either elaborated metrics not surpassing baselines, or a very recent benchmark being unable to distinguish between simple and elaborate metrics.

While this observation casts doubt on the expressiveness of the benchmark results, we still want to highlight that metrics, which were not original designed for faithfulness or in a reference free manner, like ROUGE, DiscoScore and BERTScore, can still provide some value with respect to faithfulness evaluation.

Table 7.16.: For the overall overview we assume the best metric version was selected with respect to each dataset. We use our version of DAE, QUESTEVl and SUMMAC. Balanced accuracy is reported and for enhanced comprehensibility a bucket. Bucket value is calculated by $\lfloor (accuracy - 0.5)/2 * 100 \rfloor$. The worst bucket is 0 representing the random level. The best bucket is 25 for perfect accuracy. Bucket values above the level of the best simple method (Grusky) are highlighted.

	CNN/DM		XSUM	
metric	balanced accuracy	bucket	balanced accuracy	bucket
DAE	0.630	6	0.667	8
Questeval	0.702	10	0.642	7
SummaC	0.640	7	0.635	6
ChatGPT-ZS	0.562	3	0.657	7
ChatGPT-CoT	0.543	2	0.661	8
ChatGPT-DA	0.546	2	0.548	2
ChatGPT-Star	0.535	1	0.630	6
ROUGE	0.675	8	0.536	1
Naive	0.634	6	0.587	4
Grusky	0.680	9	0.621	6
BLANC	0.681	9	0.520	1
DiscoScore	0.507	0	0.601	5
BERTScore	0.688	9	0.600	5
CTC	0.688	9	0.631	6
SGwLM	0.710	10	0.570	3
Perplexity	0.552	2	0.569	3
ESTIME	0.690	9	0.591	4
Ensemble	0.691	9	0.557	2

7.4. Overall Quality Dataset vs AGGREFACT Performance

As the ensemble did not perform well, we investigated the relation between metric performances on the overall quality dataset and AGGREFACT. Table 7.17 lists the correlation of each metric used as a feature and its balanced accuracy on the CNN/DM and XSUM data splits. The correlation coefficient between CNN/DM and XSUM accuracy is 0.087. This aligns with the observation in the preceding sections that both parts exhibit significant differences in characteristics. The correlation of performance on the overall quality dataset with CNN/DM and XSUM is 0.512 and -0.475, respectively. This might explain why the ensemble performed much better on CNN/DM than on XSUM. In conclusion, successful ensemble training necessitates a better suited training dataset.

Table 7.17.: Performance comparison, overall quality dataset vs faithfulness AGGREFACT dataset, of metrics selected as ensemble features. Per metric the correlation on the Training data set is reported, as calculated in run four (slightly deviates from other runs to randomness in the ELO calculation). Also per metric the binary accuracy on the CNN/DM and XSUM split of AGGREFACT is reported.

Metric	Version	Correlation	Balanced accuracy for CNN/DM	Balanced accuracy for XSUM
Naive	Compression	-0.28328961202731	0.485	0.587
BERTScore	Recall	0.28055883511517543	0.688	0.478
Grusky	Compression	-0.2680765482879948	0.488	0.586
Naive	Length	0.25208957758974054	0.554	0.535
ROUGE	1-Stemming	0.25139532661349334	0.629	0.461
ROUGE	1	0.2504937167050286	0.627	0.466
Naive	Length-Share	0.2422069756587334	0.634	0.500
BERTScore	F1	0.23909764870959846	0.679	0.511
DiscoScore	Sent-NN-Longformer	0.22100214024252354	0.498	0.502
DiscoScore	Sent-Entity-Longformer	0.22100214024252354	0.498	0.502
QuestEval	0.1.1-Recall	0.20795593490321437	0.622	0.499
ROUGE	Lsum-Stemming	0.20750289524153773	0.671	0.468
ROUGE	L-Stemming	0.20742905359914635	0.671	0.462
DiscoScore	Sent-NN-Truncate	0.20312366012461183	0.499	0.486
DiscoScore	Sent-Entity-Truncate	0.20312366012461183	0.499	0.486
SGwLM	ShannonScore	0.20264231801117344	0.710	0.492
ROUGE	Lsum	0.20156383540486147	0.656	0.481
ROUGE	L	0.20077201505068487	0.656	0.481
SGwLM	InfoDiff	0.1958980355057016	0.578	0.570
BLANC	Help	0.18054418241632347	0.666	0.471
ROUGE	2-Stemming	0.16490832658897708	0.675	0.535
ROUGE	2	0.15761019502446744	0.668	0.536
QuestEval	0.1.1-F-Score	0.15607580552228978	0.653	0.618
BERTScore	Precision	0.15041919663476763	0.687	0.600
Perplexity	LefthandContext-Swapped-Normal	-0.14846794607969024	0.421	0.517
QuestEval	0.2.4	0.13854513607287167	0.702	0.594
SGwLM	BlancShannon	0.1229812868352542	0.678	0.484
ESTIME	coherence	0.11335642121158568	0.606	0.523
QuestEval	0.1.1-Precision	0.11258584632573128	0.650	0.642
ESTIME	soft	0.1097938394558981	0.690	0.591
Perplexity	FullContext-SummaryOnly-Normal	0.08795990302834333	0.500	0.504
Grusky	Coverage	0.07473122612956802	0.666	0.621
SummaC	ZS	0.07371177003105017	0.640	0.539
CTC	D-CNNDM-Consistency	0.06877440300219402	0.682	0.594
ESTIME	alarms-allTokens	0.06759728216389269	0.484	0.496
CTC	D-CNNDM-Relevance	0.06655564936903537	0.669	0.590
ESTIME	alarms-notAllTokens	0.06290902694063359	0.469	0.532
Perplexity	LefthandContext-SummaryOnly-Normal	-0.061697442255971645	0.499	0.487
DAE		0.0578589399941095	0.630	0.667
Perplexity	FullContext-Normal	0.05511391453452599	0.531	0.503
DiscoScore	Focus-NN-Truncate	-0.048220763706373154	0.507	0.591
DiscoScore	Focus-ENTITY-Truncate	-0.048220763706373154	0.507	0.591
CTC	D-XSUM-Consistency	0.046320650220486946	0.688	0.631
Grusky	Density	0.031182373117535883	0.680	0.598
DiscoScore	Focus-NN-Longformer	-0.02994591350935117	0.437	0.575
DiscoScore	Focus-Entity-Longformer	-0.02994591350935117	0.437	0.575
SummaC	Conv	-0.023263442002242098	0.610	0.635
Perplexity	LefthandContext-Normal	0.0007524764573800616	0.502	0.503

8. Future Work

The fact that AGGREFACT can hardly distinguish simple baseline from elaborated metrics calls for further improvement of the benchmark. This means that more human judgment data needs to be collected. A conclusion which is in line with the assessment of other papers [15, 25, 5].

We are interested in approaches that could minimise the human labour needed to conclude the necessary data collection process.

Analysing the reason why metrics perform only slightly better than Grusky seems also very promising. One possible approach would be investigating the agreement rates of metrics and Grusky [16]. This could provide valuable information on how to improve metrics and AGGREFACT.

One potential approach to generated summary data particularly hard to handle by metrics could be to abuse the overfitting problem of RL rewards based on metrics. This is kind of inspired by the approach of Reinforcement Learning from Human Feedback (RLfHF). Agents are trained until they discovers the weak points of each a metric, these are then rated by humans to form or supplement a benchmark.

As we have elaborated that rewards, models and summaries have to be assessed with respect to faithfulness and overall quality, we think more rated data for overall human quality is needed as well.

In the future, it may be possible to enhance the ensemble by training on the entire dataset, thus addressing the disparity between training and validation performance. However the data set seems to match only the characteristics of CNN/DM, therefore a more diverse dataset needs to be collected. One potential addition could be the dataset collected by Vasilyev et al. [68]. Investigating into whether additional synthetic training data would be beneficial seems also promising.

In the future, we are also interested in comparing a reward trained from scratch using human feedback with one trained based on features. We hypothesize that ensembling could potentially reduce the data requirement.

As the metrics perform so differently on CNN/DM and XSUM it might also be promising to add a classifier as feature into the ensemble which detects the dataset characteristic. This way the ensemble could decide which metrics to rely on for different kinds of data.

9. Conclusion

Faithfulness is a central quality aspect for summarization. However assessing it in isolation offers limited benefit as isolated faithfulness has a trivial solution.

Reinforcement Learning is seen as promising research direction to develop improved summarizers as it is less data dependent and circumvents the problem of exposure bias. Therefore this thesis elaborates different RL approaches and the importance of a good reward. Relying on human data can provide good alignment but is very resource intensive. Incorporating metrics is cheaper and therefore more accessible. However good metrics are lacking. Sub-optimal metrics in rewards push the risk of overfitting, which can currently only be solved by human evaluation. Furthermore benchmarks to determine metric performance are also criticised. The main bottleneck for benchmark creation is again the high labour requirement for human judgments.

We selected AGGREFACT as very recent and comprehensive benchmark and evaluated 17 metrics and their sub-versions: Grusky, Naive, ROUGE, DAE, QuestEval, SummaC, DiscoScore, ESTIME, SGwLM, BLANC, Perplexity, BERTScore, CTC and ChatGPT-ZS, -CoT, -DA and -Star.

First of all our results reproduced trends reported in the AGGREFACT paper by Tang et al. [47] with one exception: ChatGPT-CoT. ChatGPT provides free text ratings which need to be converted using a heuristic. These are lacking in the original papers proposing the ChatGPT-based metrics and the AGGREFACT paper. Relying on a simple metric aligns our results to Tang's. We investigated the influence of different heuristics and found that one, which relies on ChatGPT again, to convert the answer to a fixed format, improves the performance of CoT significantly. Given the improved heuristics the performance of CoT does not longer align with the results presented by Tang, but it aligns with the original paper by Luo et al. [128].

A shocking result is that the best simple metric, which we added as baseline and relies on word-matches, is nearly as good as the best intricate metric. Either metrics did not really progress or the AGGREFACT benchmark is unable to point out the differences. To figure out which option applies requires more research, however both are crucial.

Motivated by the lack of proper metrics, which could be used as reward building blocks, and the tension between overall quality and faithfulness, we developed an ensemble metric which incorporated most of the presented metrics as features. It is trained on an aggregated dataset composed of data from Goyal [30] and Stiennon [45]. These are tailored for overall summary quality. The hypothesis is that improved overall quality capabilities of a metric will eventually align with improved faithfulness assessment, as faithfulness is part of overall quality. The training is carried out using an EA, as we directly optimise for correlation which is not differentiable. The training was concluded successfully but did not translate to

faithfulness. We assume the training dataset was insufficient. The analysis shows that the CNN/DM and XSUM part of AGGREFACT occupy very different data set characteristics. Good performance on one does hardly translate to the other. However our training dataset did only align with CNN/DM split.

Therefore we conclude that better data is required for optimal metric assessment and ensemble training. However, for now we present an overview of RL approaches and metrics.

A. Metric details

A.1. ChatGPT Prompts

These are the prompts proposed by Luo [128] and Wang [129], used in AGGREFACT [47] and our evaluation. *[Article]* and *[Summary]* are supposed to be replaced with the article and summary of interest.

A.1.1. ZS

Decide if the following summary is consistent with the corresponding article. Note that consistency means all information in the summary is supported by the article.
Article: [Article]
Summary: [Summary]
Answer (yes or no):

A.1.2. CoT

Decide if the following summary is consistent with the corresponding article. Note that consistency means all information in the summary is supported by the article.
Article: [Article]
Summary: [Summary]
Explain your reasoning step by step then answer (yes or no) the question:

A.1.3. Da

Score the following news summarization given the corresponding news with respect to consistency on a continuous scale from 0 to 100, where a score of zero means "inconsistency" and score of one hundred means "perfect consistency". Note that consistency measures whether the facts in the summary are consistent with the facts in the original article. Consider whether the summary does reproduce all facts accurately and does not make up untrue information.
Article: [Article]
Summary: [Summary]
Scores:

A.1.4. Star

Score the following news summarization given the corresponding news with respect to consistency with one to five stars, where one star means "inconsistency" and five stars means "perfect consistency".

Note that consistency measures whether the facts in the summary are consistent with the facts in the original article. Consider whether the summary does reproduce all facts accurately and does not make up untrue information.
Article: [Article]
Summary: [Summary]
Stars:

List of Figures

List of Tables

Acronyms

ATS *Automatic Text Summarization. 1*

DAE *Dependency Arc Entailment. iii, iv, 13, 15, 16, 45, 47*

DL *Deep Learning. 6, 13*

DS *DiscoScore. iii, iv, 13, 17, 18, 38, 42, 45, 47*

EA *Evolutionary Algorithm. iii, 8, 29, 30, 47*

GAN *Generative Adversarial Networks. vi, 9, 10*

IE *Information Extraction. 5, 6, 16, 17*

LLM *Large Language Model. 18–20*

LLMs *Large Language Models. 6, 9*

NLI *Natural Language Inference (also known as Textual Entailment). 5, 6, 16*

QA *Question Answering. 5, 6, 16*

RL *Reinforcement Learning. iii–vi, 2, 3, 8–12, 25, 27, 29, 46–48*

RLfHF *Reinforcement Learning from Human Feedback. vi, 9, 10, 46*

SGwLM *Shannon Game with Language Model. iii, iv, 17, 18, 39, 47*

SL *Supervised Learning. iv, vi, 9, 11, 29*

Bibliography

[1] O. Vasilyev and J. Bohannon. "Is human scoring the best criteria for summary evaluation?" In: (Dec. 29, 2020). arXiv: 2012.14602 [cs.CL].

[2] M. Bhandari, P. Gour, A. Ashfaq, P. Liu, and G. Neubig. "Re-evaluating Evaluation in Text Summarization". In: (Oct. 14, 2020). DOI: 10.18653/v1/2020.emnlp-main.751. arXiv: 2010.07100 [cs.CL].

[3] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir. "A Survey of Automatic Text Summarization: Progress, Process and Challenges". In: *IEEE Access* 9 (2021), pp. 156043–156070. DOI: 10.1109/access.2021.3129786.

[4] D. Suleiman and A. Awajan. "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges". In: *Mathematical Problems in Engineering* 2020 (Aug. 2020), pp. 1–29. DOI: 10.1155/2020/9365340.

[5] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, and Y. Koreeda. "Holistic Evaluation of Language Models". In: (Nov. 16, 2022). arXiv: 2211.09110 [cs.CL].

[6] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher. "Neural Text Summarization: A Critical Evaluation". 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/d19-1051.

[7] T. Falke, L. F. R. Ribeiro, P. A. Utama, I. Dagan, and I. Gurevych. "Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/p19-1213.

[8] Z. Cao, F. Wei, W. Li, and S. Li. "Faithful to the Original: Fact Aware Neural Abstractive Summarization". In: (Nov. 13, 2017). arXiv: 1711.04434 [cs.IR].

[9] S. Gabriel, A. Celikyilmaz, R. Jha, Y. Choi, and J. Gao. "GO FIGURE: A Meta Evaluation of Factuality in Summarization". In: (Oct. 24, 2020). arXiv: 2010.12834 [cs.CL].

[10] C. Zhou, G. Neubig, J. Gu, M. Diab, F. Guzmán, L. Zettlemoyer, and M. Ghazvininejad. "Detecting Hallucinated Content in Conditional Neural Sequence Generation". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.findings-acl.120.

[11] C. Zhu, W. Hinthorn, R. Xu, Q. Zeng, M. Zeng, X. Huang, and M. Jiang. "Enhancing Factual Consistency of Abstractive Summarization". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.naacl-main.58.

[12] T. Goyal and G. Durrett. "Annotating and Modeling Fine-grained Factuality in Summarization". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.naacl-main.114.

[13] M. Zhang, G. Zhou, W. Yu, N. Huang, and W. Liu. "A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning". In: *Computational Intelligence and Neuroscience* 2022 (Aug. 2022). Ed. by A. Hosovsky, pp. 1–21. DOI: 10.1155/2022/7132226.

[14] H. Lin and V. Ng. "Abstractive Summarization: A Survey of the State of the Art". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), pp. 9815–9822. DOI: 10.1609/aaai.v33i01.33019815.

[15] S. Gehrmann, E. Clark, and T. Sellam. "Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text". In: (Feb. 14, 2022). arXiv: 2202.06935 [cs.CL].

[16] E. Durmus, F. Ladhak, and T. Hashimoto. "Spurious Correlations in Reference-Free Evaluation of Text Generation". In: (Apr. 21, 2022). arXiv: 2204.09890 [cs.CL].

[17] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, Mar. 2021, pp. 610–623. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

[18] J. Kasai, K. Sakaguchi, R. L. Bras, L. Dunagan, J. Morrison, A. Fabbri, Y. Choi, and N. Smith. "Bidimensional Leaderboards: Generate and Evaluate Language Hand in Hand". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022. DOI: 10.18653/v1/2022.naacl-main.259.

[19] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung. "Survey of Hallucination in Natural Language Generation". In: (Feb. 8, 2022). arXiv: 2202.03629 [cs.CL].

[20] M. Cao. "A Survey on Neural Abstractive Summarization Methods and Factual Consistency of Summarization". In: (Apr. 20, 2022). arXiv: 2204.09519 [cs.CL].

[21] W. Chen, P. Li, and I. King. "A Training-free and Reference-free Summarization Evaluation Metric via Centrality-weighted Relevance and Self-referenced Redundancy". In: (June 26, 2021). arXiv: 2106.13945 [cs.CL].

[22] Y. Liu, A. R. Fabbri, P. Liu, Y. Zhao, L. Nan, R. Han, S. Han, S. Joty, C.-S. Wu, C. Xiong, and D. Radev. "Revisiting the Gold Standard: Grounding Summarization Evaluation with Robust Human Evaluation". In: (Dec. 15, 2022). arXiv: 2212.07981 [cs.CL].

[23] O. Honovich, R. Aharoni, J. Herzig, H. Taitelbaum, D. Kukliansy, V. Cohen, T. Scialom, I. Szpektor, A. Hassidim, and Y. Matias. "TRUE: Re-evaluating Factual Consistency Evaluation". In: (Apr. 11, 2022). arXiv: 2204.04991 [cs.CL].

[24] T. Sellam, D. Das, and A. Parikh. "BLEURT: Learning Robust Metrics for Text Generation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.704.

[25] M. Bhandari, P. N. Gour, A. Ashfaq, and P. Liu. "Metrics also Disagree in the Low Scoring Range: Revisiting Summarization Evaluation Metrics". In: *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020. DOI: 10.18653/v1/2020.coling-main.501.

[26] F. Koto, T. Baldwin, and J. H. Lau. "FFCI: A Framework for Interpretable Automatic Evaluation of Summarization". In: (Feb. 28, 2022). arXiv: 2011.13662 [cs.CL].

[27] P. K. Choubey, A. R. Fabbri, J. Vig, C.-S. Wu, W. Liu, and N. F. Rajani. "CaPE: Contrastive Parameter Ensembling for Reducing Hallucination in Abstractive Summarization". In: (Oct. 14, 2021). arXiv: 2110.07166 [cs.CL].

[28] Y. Huang, X. Feng, X. Feng, and B. Qin. "The Factual Inconsistency Problem in Abstractive Text Summarization: A Survey". In: (Apr. 30, 2021). arXiv: 2104.14839 [cs.CL].

[29] T. Fischer, S. Remus, and C. Biemann. "Measuring Faithfulness of Abstractive Summaries". In: *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*. Potsdam, Germany: KONVENS 2022 Organizers, Sept. 15, 2022, pp. 63–73. URL: https://aclanthology.org/2022.konvens-1.8 (visited on 10/01/2022).

[30] T. Goyal, J. J. Li, and G. Durrett. "News Summarization and Evaluation in the Era of GPT-3". In: (Sept. 26, 2022). arXiv: 2209.12356 [cs.CL].

[31] C.-Y. Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[32] N. Egan, O. Vasilyev, and J. Bohannon. "Play the Shannon Game With Language Models: A Human-Free Approach to Summary Evaluation". In: (Mar. 19, 2021). arXiv: 2103.10918 [cs.CL].

[33] A. Alomari, N. Idris, A. Q. M. Sabri, and I. Alsmadi. "Deep reinforcement and transfer learning for abstractive text summarization: A review". In: *Computer Speech & Language* 71 (Jan. 2022), p. 101276. DOI: 10.1016/j.csl.2021.101276.

[34] H. Y. Koh, J. Ju, M. Liu, and S. Pan. "An Empirical Survey on Long Document Summarization: Datasets, Models and Metrics". In: (July 3, 2022). DOI: 10.1145/3545176. arXiv: 2207.00939 [cs.CL].

[35] C. Garbacea and Q. Mei. "Neural Language Generation: Formulation, Methods, and Evaluation". In: (July 31, 2020). arXiv: 2007.15780 [cs.CL].

[36] P. A. Rankel, J. M. Conroy, and J. D. Schlesinger. "Better Metrics to Automatically Predict the Quality of a Text Summary". In: *Algorithms* 5.4 (Sept. 2012), pp. 398–420. DOI: 10.3390/a5040398.

[37] F. Retkowski. "Reinforcement Learning for Sequence-to-Sequence Dialogue Systems". MA thesis. Karlsruhe Institute of Technology, 2020. URL: https://www.semanticscholar.org/paper/Reinforcement-Learning-for-Sequence-to-Sequence-Retkowski/94486ed250a0a679a6487217aee2d1feaa642fac (visited on 08/03/2022).

[38] J. Parnell, I. J. Unanue, and M. Piccardi. "RewardsOfSum: Exploring Reinforcement Learning Rewards for Summarisation". In: (June 8, 2021). arXiv: 2106.04080 [cs.CL].

[39] Y. Chen, L. Wu, and M. J. Zaki. "Reinforcement Learning Based Graph-to-Sequence Model for Natural Question Generation". In: (Aug. 14, 2019). arXiv: 1908.04942 [cs.CL].

[40] H. Jang and W. Kim. "Reinforced Abstractive Text Summarization With Semantic Added Reward". In: *IEEE Access* 9 (2021), pp. 103804–103810. DOI: 10.1109/access.2021.3097087.

[41] G. Neubig. *CMU Neural Nets for NLP 2021 (14): Margin-based and Reinforcement Learning for Structured Prediction*. Mar. 22, 2021. URL: https://www.youtube.com/watch?v=3YCb-F4pS4E&list=PL8PYTP1V4I8AkaHEJ7lOOrlex-pcxS-XV&index=15.

[42] R. Paulus, C. Xiong, and R. Socher. "A Deep Reinforced Model for Abstractive Summarization". In: (May 11, 2017). arXiv: 1705.04304 [cs.CL].

[43] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy. "Neural Abstractive Text Summarization with Sequence-to-Sequence Models". In: *ACM/IMS Transactions on Data Science* 2.1 (Feb. 2021), pp. 1–37. DOI: 10.1145/3419106. arXiv: 1812.02303 [cs.CL].

[44] T. Scialom, S. Lamprier, B. Piwowarski, and J. Staiano. "Answers Unite! Unsupervised Metrics for Reinforced Summarization Models". 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/d19-1320.

[45] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano. "Learning to summarize from human feedback". In: (Sept. 2, 2020). arXiv: 2009.01325 [cs.CL].

[46] C. An, M. Zhong, Z. Wu, Q. Zhu, X. Huang, and X. Qiu. "COLO: A Contrastive Learning based Re-ranking Framework for One-Stage Summarization". In: (Sept. 29, 2022). arXiv: 2209.14569 [cs.CL].

[47] L. Tang, T. Goyal, A. R. Fabbri, P. Laban, J. Xu, S. Yahvuz, W. Kryściński, J. F. Rousseau, and G. Durrett. "Understanding Factual Errors in Summarization: Errors, Summarizers, Datasets, Error Detectors". In: (May 25, 2022). arXiv: 2205.12854 [cs.CL].

[48] H. Kane, M. Y. Kocyigit, A. Abdalla, P. Ajanoh, and M. Coulibali. "NUBIA: NeUral Based Interchangeability Assessor for Text Generation". In: (Apr. 30, 2020). arXiv: 2004.14667 [cs.CL].

[49] P. Manakul and M. J. F. Gales. "Podcast Summary Assessment: A Resource for Evaluating Summary Assessment Methods". In: (Aug. 28, 2022). arXiv: 2208.13265 [cs.CL].

[50] Y.-C. Chen and M. Bansal. "Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting". In: (May 28, 2018). DOI: 10.18653/v1/p18-1063. arXiv: 1805.11080 [cs.CL].

[51] Y. Zhang, A. Ni, Z. Mao, C. H. Wu, C. Zhu, B. Deb, A. H. Awadallah, D. Radev, and R. Zhang. "Summ^N: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents". In: (Oct. 16, 2021). arXiv: 2110.10150 [cs.CL].

[52] M. Peyrard, T. Botschen, and I. Gurevych. "Learning to Score System Summaries for Better Content Selection Evaluation." In: *Proceedings of the Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/w17-4510.

[53] D. O. Cajueiro, A. G. Nery, I. Tavares, M. K. D. Melo, S. A. dos Reis, L. Weigang, and V. R. R. Celestino. "A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding". In: (Jan. 4, 2023). arXiv: 2301.03403 [cs.CL].

[54] P. Verma and A. Verma. "A Review on Text Summarization Techniques". In: *Journal of scientific research* 64.01 (2020), pp. 251–257. DOI: 10.37398/jsr.2020.640148. URL: https://www.researchgate.net/publication/339146200_A_Review_on_Text_Summarization_Techniques.

[55] N. Babakov, D. Dale, V. Logacheva, and A. Panchenko. "A large-scale computational study of content preservation measures for text style transfer and paraphrase generation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, 2022. DOI: 10.18653/v1/2022.acl-srw.23.

[56] M. Gao and X. Wan. "DialSummEval: Revisiting Summarization Evaluation for Dialogues". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022. DOI: 10.18653/v1/2022.naacl-main.418.

[57] K. Arumae and F. Liu. "Reinforced Extractive Summarization with Question-Focused Rewards". In: *Proceedings of ACL 2018, Student Research Workshop*. Association for Computational Linguistics, 2018.

[58] W. Fabbri, B. Kryscinski, C. Mccann, R. Xiong, D. Socher, and Radev. "SummEval: Re-evaluating Summarization Evaluation". In: *Transactions of the Association for Computational Linguistics* 9.0 (2021), pp. 391–409. ISSN: 2307-387X. URL: https://transacl.org/ojs/index.php/tacl/article/view/2563 (visited on 09/30/2022).

[59] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. Christiano. "Recursively Summarizing Books with Human Feedback". In: (Sept. 22, 2021). arXiv: 2109.10862 [cs.CL].

[60] S. Galeshchuk. *"Can you believe what you summed up?" or how to evaluate a text summarization model*. Mar. 23, 2022. URL: https://medium.com/@svitlana.galeshchuk/can-you-believe-what-you-read-or-how-well-your-model-summarizes-a-text-8301e54d3ba5.

[61] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed. "Automatic text summarization: A comprehensive survey". In: *Expert Systems with Applications* 165 (Mar. 2021), p. 113679. DOI: 10.1016/j.eswa.2020.113679.

[62] M. Deng, B. Tan, Z. Liu, E. Xing, and Z. Hu. "Compression, Transduction, and Creation: A Unified Framework for Evaluating Natural Language Generation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.599.

[63] D. Jin, Z. Jin, Z. Hu, O. Vechtomova, and R. Mihalcea. "Deep Learning for Text Style Transfer: A Survey". In: (Nov. 1, 2020). arXiv: 2011.00416 [cs.CL].

[64] H. Hardy, S. Narayan, and A. Vlachos. "HighRES: Highlight-based Reference-less Evaluation of Summarization". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/p19-1330.

[65] H. Rashkin, V. Nikolaev, M. Lamm, L. Aroyo, M. Collins, D. Das, S. Petrov, G. S. Tomar, I. Turc, and D. Reitter. "Measuring Attribution in Natural Language Generation Models". In: (Dec. 23, 2021). arXiv: 2112.12870 [cs.CL].

[66] A. Pagnoni, V. Balachandran, and Y. Tsvetkov. "Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics". In: (Apr. 27, 2021). arXiv: 2104.13346 [cs.CL].

[67] S. Zhang and M. Bansal. "Finding a Balanced Degree of Automation for Summary Evaluation". In: (Sept. 23, 2021). arXiv: 2109.11503 [cs.CL].

[68] O. Vasilyev, V. Dharnidharka, and J. Bohannon. "Fill in the BLANC: Human-free quality estimation of document summaries". In: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.eval4nlp-1.2.

[69] Y. Graham. "Re-evaluating Automatic Summarization with BLEU and 192 Shades of ROUGE". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015. DOI: 10.18653/v1/d15-1013.

[70] R. K. Amplayo and M. Lapata. "Unsupervised Opinion Summarization with Noising and Denoising". In: (Apr. 21, 2020). arXiv: 2004.10150 [cs.CL].

[71] N. Burton, M. Burton, D. Rigby, C. A. M. Sutherland, and G. Rhodes. "Best-worst scaling improves measurement of first impressions". In: *Cognitive Research: Principles and Implications* 4.1 (Sept. 2019). DOI: 10.1186/s41235-019-0183-2.

[72] S. Kiritchenko and S. Mohammad. "Best-Worst Scaling More Reliable than Rating Scales: A Case Study on Sentiment Intensity Annotation". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/p17-2074.

[73] G. Hollis and C. Westbury. "When is best-worst best? A comparison of best-worst scaling, numeric estimation, and rating scales for collection of semantic norms". In: *Behavior Research Methods* 50.1 (Jan. 2018), pp. 115–133. DOI: 10.3758/s13428-017-1009-0.

[74] G. Hollis. "Scoring best-worst data in unbalanced many-item designs, with applications to crowdsourcing semantic judgments". In: *Behavior Research Methods* 50.2 (May 2017), pp. 711–729. DOI: 10.3758/s13428-017-0898-2.

[75] H. Lee, K. M. Yoo, J. Park, H. Lee, and K. Jung. "Masked Summarization to Generate Factually Inconsistent Summaries for Improved Factual Consistency Checking". In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, 2022. DOI: 10.18653/v1/2022.findings-naacl.76.

[76] Z. Zeng, J. Chen, W. Xu, and L. Li. "Gradient-Based Adversarial Factual Consistency Evaluation for Abstractive Summarization". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.337.

[77] S. Zhang, J. Niu, and C. Wei. "Fine-grained Factual Consistency Assessment for Abstractive Summarization Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.9.

[78] W. Wu, W. Li, X. Xiao, J. Liu, S. Li, and Y. Lv. "WeCheck: Strong Factual Consistency Checker via Weakly Supervised Learning". In: (Dec. 20, 2022). arXiv: 2212.10057 [cs.CL].

[79] Y. Zhang, Y. Zhang, and C. D. Manning. "A Close Examination of Factual Correctness Evaluation in Abstractive Summarization". In: 2020. URL: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/reports/custom/report53.pdf.

[80] D. Tam, A. Mascarenhas, S. Zhang, S. Kwan, M. Bansal, and C. Raffel. "Evaluating the Factual Consistency of Large Language Models Through Summarization". In: (Nov. 15, 2022). arXiv: 2211.08412 [cs.CL].

[81] A. R. Fabbri, C.-S. Wu, W. Liu, and C. Xiong. "QAFactEval: Improved QA-Based Factual Consistency Evaluation for Summarization". In: (Dec. 16, 2021). arXiv: 2112.08542 [cs.CL].

[82] R. Aralikatte, S. Narayan, J. Maynez, S. Rothe, and R. McDonald. "Focus Attention: Promoting Faithfulness and Diversity in Summarization". In: (May 25, 2021). arXiv: 2105.11921 [cs.CL].

[83] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. "BERTScore: Evaluating Text Generation with BERT". In: (Apr. 21, 2019). arXiv: 1904.09675 [cs.CL].

[84] R. Kamoi, T. Goyal, and G. Durrett. "Shortcomings of Question Answering Based Factuality Frameworks for Error Localization". In: (Oct. 13, 2022). arXiv: 2210.06748 [cs.CL].

[85] L. Ribeiro, M. Liu, I. Gurevych, M. Dreyer, and M. Bansal. "FactGraph: Evaluating Factuality in Summarization with Semantic Graph Representations". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022. DOI: 10.18653/v1/2022.naacl-main.236.

[86] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst. "SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization". In: (Nov. 18, 2021). arXiv: 2111.09525 [cs.CL].

[87] A. Mishra, D. Patel, A. Vijayakumar, X. Li, P. Kapanipathi, and K. Talamadupula. "Looking Beyond Sentence-Level Natural Language Inference for Downstream Tasks". In: (Sept. 18, 2020). arXiv: 2009.09099 [cs.CL].

[88] W. Yin, D. Radev, and C. Xiong. "DocNLI: A Large-scale Dataset for Document-level Natural Language Inference". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.findings-acl.435.

[89] M. Barrantes, B. Herudek, and R. Wang. "Adversarial NLI for Factual Correctness in Text Summarisation Models". In: (May 24, 2020). arXiv: 2005.11739 [cs.CL].

[90] P. A. Utama, J. Bambrick, N. S. Moosavi, and I. Gurevych. "Falsesum: Generating Document-level NLI Examples for Recognizing Factual Inconsistency in Summarization". In: (May 12, 2022). arXiv: 2205.06009 [cs.CL].

[91] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. "On Faithfulness and Factuality in Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.173.

[92] Y. Xie, F. Sun, Y. Deng, Y. Li, and B. Ding. "Factual Consistency Evaluation for Text Summarization via Counterfactual Estimation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.findings-emnlp.10.

[93] H. Jin, Y. Cao, T. Wang, X. Xing, and X. Wan. "Recent advances of neural text generation: Core tasks, datasets, models and challenges". In: *Science China Technological Sciences* 63.10 (Sept. 2020), pp. 1990–2010. DOI: 10.1007/s11431-020-1622-y. (Visited on 09/13/2022).

[94] M. Cao, Y. Dong, and J. Cheung. "Hallucinated but Factual! Inspecting the Factuality of Hallucinations in Abstractive Summarization". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2022. DOI: 10.18653/v1/2022.acl-long.236.

[95] S. Son, J. Park, J.-i. Hwang, J. Lee, H. Noh, and Y. Lee. "HaRiM$^+$: Evaluating Summary Quality with Hallucination Risk". In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (AACL-IJCNLP 2022), pages 895-924* (Nov. 22, 2022). arXiv: 2211.12118 [cs.CL].

[96] P. Colombo, G. Staerman, C. Clavel, and P. Piantanida. "Automatic Text Evaluation through the Lens of Wasserstein Barycenters". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.817.

[97] F. S. Bao, R. Tu, and G. Luo. "DocAsRef: A Pilot Empirical Study on Repurposing Reference-Based Summary Quality Metrics Reference-Freely". In: (Dec. 20, 2022). arXiv: 2212.10013 [cs.AI].

[98] S. Chaudhury, S. Swaminathan, C. Gunasekara, M. Crouse, S. Ravishankar, D. Kimura, K. Murugesan, R. F. Astudillo, T. Naseem, P. Kapanipathi, and A. G. Gray. "X-FACTOR: A Cross-metric Evaluation of Factual Correctness in Abstractive Summarization". In: *Conference on Empirical Methods in Natural Language Processing*. 2022. URL: https://www.semanticscholar.org/paper/X-FACTOR:-A-Cross-metric-Evaluation-of-Factual-in-Chaudhury-Swaminathan/e59cf68d162ddfe7052316548515349e38720160.

[99] D. Hyun, X. Wang, C. Park, X. Xie, and H. Yu. "Generating Multiple-Length Summaries via Reinforcement Learning for Unsupervised Sentence Summarization". In: (Dec. 21, 2022). arXiv: 2212.10843 [cs.CL].

[100] A. Wang, R. Y. Pang, A. Chen, J. Phang, and S. R. Bowman. "SQuALITY: Building a Long-Document Summarization Dataset the Hard Way". In: (May 23, 2022). arXiv: 2205.11465 [cs.CL].

[101] M. Xia, E. Kochmar, and T. Briscoe. "Automatic learner summary assessment for reading comprehension". In: *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/n19-1261.

[102] D. Deutsch, R. Dror, and D. Roth. "Re-Examining System-Level Correlations of Automatic Summarization Evaluation Metrics". In: (Apr. 21, 2022). arXiv: 2204.10216 [cs.CL].

[103] M. Kaster, W. Zhao, and S. Eger. "Global Explainability of BERT-Based Evaluation Metrics by Disentangling along Linguistic Factors". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.701.

[104] M. Peyrard. "Studying Summarization Evaluation Metrics in the Appropriate Scoring Range". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/p19-1502.

[105] A. Louis and A. Nenkova. "Automatically Assessing Machine Summary Content Without a Gold Standard". In: *Computational Linguistics* 39.2 (June 2013), pp. 267–300. DOI: 10.1162/coli_a_00123.

[106] D. Deutsch, R. Dror, and D. Roth. "A Statistical Analysis of Summarization Evaluation Metrics Using Resampling Methods". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 1132–1146. DOI: 10.1162/tacl_a_00417.

[107] O. Vasilyev and J. Bohannon. "ESTIME: Estimation of Summary-to-Text Inconsistency by Mismatched Embeddings". In: *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.eval4nlp-1.10.

[108] L. Ma, S. Cao, R. L. L. IV, D. Lu, S. Ran, K. Zhang, J. Tetreault, A. Cahill, and A. Jaimes. "BUMP: A Benchmark of Unfaithful Minimal Pairs for Meta-Evaluation of Faithfulness Metrics". In: (Dec. 20, 2022). arXiv: 2212.09955 [cs.CL].

[109] Y. Chen and S. Eger. "MENLI: Robust Evaluation Metrics from Natural Language Inference". In: (Aug. 15, 2022). arXiv: 2208.07316 [cs.CL].

[110] Y.-x. He, D.-x. Liu, D.-h. Ji, H. Yang, and C. Teng. "MSBGA: A Multi-Document Summarization System Based on Genetic Algorithm". In: *2006 International Conference on Machine Learning and Cybernetics*. IEEE, 2006. DOI: 10.1109/icmlc.2006.258921.

[111] Wikipedia. *Evolutionary algorithm*. Aug. 6, 2023. URL: https://en.wikipedia.org/wiki/Evolutionary_algorithm (visited on 08/06/2023).

[112] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. "Deep reinforcement learning from human preferences". In: (June 12, 2017). arXiv: 1706.03741 [stat.ML].

[113] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal Policy Optimization Algorithms". In: (July 20, 2017). arXiv: 1707.06347 [cs.LG].

[114] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. "Training language models to follow instructions with human feedback". In: (Mar. 4, 2022). arXiv: 2203.02155 [cs.CL].

[115] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman. "WebGPT: Browser-assisted question-answering with human feedback". In: (Dec. 17, 2021). arXiv: 2112.09332 [cs.CL].

[116] F. Böhm, Y. Gao, C. M. Meyer, O. Shapira, I. Dagan, and I. Gurevych. "Better Rewards Yield Better Summaries: Learning to Summarise Without References". In: (Sept. 3, 2019). arXiv: 1909.01214 [cs.CL].

[117] P. Li, L. Bing, and W. Lam. "Actor-Critic based Training Framework for Abstractive Summarization". In: (Mar. 28, 2018). arXiv: 1803.11070 [cs.CL].

[118] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators". In: (Mar. 23, 2020). arXiv: 2003.10555 [cs.CL].

[119] K. Arumae and F. Liu. "Guiding Extractive Summarization with Question-Answering Rewards". In: (Apr. 4, 2019). arXiv: 1904.02321 [cs.CL].

[120] C. Gunasekara, G. Feigenblat, B. Sznajder, R. Aharonov, and S. Joshi. "Using Question Answering Rewards to Improve Abstractive Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.findings-emnlp.47.

[121] S. Zhang, D. Wan, and M. Bansal. "Extractive is not Faithful: An Investigation of Broad Unfaithfulness Problems in Extractive Summarization". In: (Sept. 8, 2022). arXiv: 2209.03549 [cs.CL].

[122] M. Grusky, M. Naaman, and Y. Artzi. "Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/n18-1065.

[123] T. Scialom, P.-A. Dray, S. Lamprier, B. Piwowarski, J. Staiano, A. Wang, and P. Gallinari. "QuestEval: Summarization Asks for Fact-based Evaluation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.529.

[124] W. Zhao, M. Strube, and S. Eger. "DiscoScore: Evaluating Text Generation with BERT and Discourse Coherence". In: (Jan. 26, 2022). arXiv: 2201.11176 [cs.CL].

[125] I. Beltagy, M. E. Peters, and A. Cohan. "Longformer: The Long-Document Transformer". In: (Apr. 10, 2020). arXiv: 2004.05150 [cs.CL].

[126] O. Vasilyev and J. Bohannon. "Consistency and Coherence from Points of Contextual Similarity". In: (Dec. 22, 2021). arXiv: 2112.11638 [cs.CL].

[127] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: (Oct. 11, 2018). arXiv: 1810.04805 [cs.CL].

[128] Z. Luo, Q. Xie, and S. Ananiadou. "ChatGPT as a Factual Inconsistency Evaluator for Text Summarization". In: (Mar. 27, 2023). arXiv: 2303.15621 [cs.CL].

[129] J. Wang, Y. Liang, F. Meng, Z. Sun, H. Shi, Z. Li, J. Xu, J. Qu, and J. Zhou. "Is ChatGPT a Good NLG Evaluator? A Preliminary Study". In: (Mar. 7, 2023). arXiv: 2303.04048 [cs.CL].

[130] W. Li, W. Wu, M. Chen, J. Liu, X. Xiao, and H. Wu. "Faithfulness in Natural Language Generation: A Systematic Survey of Analysis, Evaluation and Optimization Methods". In: (Mar. 10, 2022). arXiv: 2203.05227 [cs.CL].

[131] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. "Teaching Machines to Read and Comprehend". In: (June 10, 2015). arXiv: 1506.03340 [cs.CL].

[132] S. Narayan, S. B. Cohen, and M. Lapata. "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/d18-1206.

[133] K.-H. Huang, S. Singh, X. Ma, W. Xiao, F. Nan, N. Dingwall, W. Y. Wang, and K. McKeown. "SWING: Balancing Coverage and Faithfulness for Dialogue Summarization". In: (Jan. 25, 2023). arXiv: 2301.10483 [cs.CL].