

Evaluation of Personalized Summarization

by

VANSH RAHUL BHANJIBHAI
202111035

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

INFORMATION AND COMMUNICATION TECHNOLOGY

to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

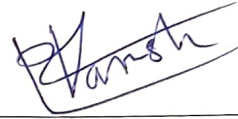


June, 2023

Declaration

I hereby declare that

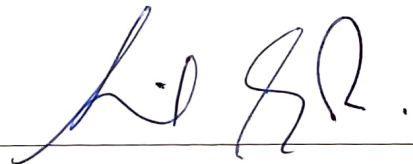
- i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,
- ii) due acknowledgment has been made in the text to all the reference material used.



Vansh Rahul Bhanjibhai

Certificate

This is to certify that the thesis work entitled ' Evaluation of Personalized Summarization' has been carried out by Vansh Rahul Bhanjibhai for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.



Prof. Sourish Dasgupta
Thesis Supervisor

Acknowledgments

I would like to express my sincere gratitude to my thesis mentor, Professor Sourish Dasgupta, for his unwavering guidance and support throughout the year. Additionally, I would like to extend my gratitude to Professor Tanmoy Chakraborty for his collaboration on this research. Their expertise and willingness to help have been instrumental in shaping my thesis work. Their constructive feedback, insightful suggestions, and encouragement have motivated me to push myself beyond my limits and achieve the best possible outcome. Thank you for your valuable contribution, patience, and unwavering support.

Contents

| | |
|---|------------|
| Abstract | v |
| List of Tables | vi |
| List of Figures | vii |
| 1 Introduction | 1 |
| 1.1 Problem statement | 1 |
| 1.2 Current work | 2 |
| 1.3 Challenges | 3 |
| 1.4 Scope of research | 4 |
| 2 Related work | 5 |
| 2.1 Summarization Model | 5 |
| 2.1.1 Personalized summarization models | 5 |
| 2.1.2 Non Personalized summarization models | 7 |
| 2.2 Summarization Evaluation | 10 |
| 2.2.1 Accuracy | 11 |
| 2.2.2 Quality | 12 |
| 3 Degree of personalization | 13 |
| 3.1 Defining Insensitivity to subjectivity | 13 |
| 4 Measuring Degree of Personalization | 16 |
| 4.1 Deviation | 16 |
| 4.2 Effective Degree of Insensitivity w.r.t subjectivity (EGISES) | 18 |
| 5 Case study on PENS dataset and framework | 20 |
| 5.1 PENS Dataset | 20 |
| 5.2 PENS framework | 21 |
| 5.3 BRIO (Bringing Order to Abstractive Summarization) | 24 |

| | | |
|----------|--|-----------|
| 5.4 | Experimental setup to compute EGISES on PENS dataset | 25 |
| 5.4.1 | OOV Handling | 29 |
| 5.5 | Results | 31 |
| 6 | Robustness of EGISES | 33 |
| 6.1 | Correlation between ROUG- L Dev and DINS Dev | 33 |
| 6.2 | Correlation between ROUGE L and Personalized ROUGE | 36 |
| 7 | Conclusions and future direction | 39 |
| | References | 44 |

Abstract

This research aims to address the limitations in evaluating the personalization of a summarizer model solely based on its accuracy. Current accuracy-based measures, such as ROUGE, fail to consider subjectivity when evaluating personalized summarization. To overcome this, we introduce a novel metric called EGISES, which evaluates the degree of personalization by taking into account both the user profile and the model generated summary. Additionally, we propose P-ROUGE, a novel metric that combines accuracy and the degree of personalization. We conduct a comprehensive analysis to establish the consistency and reliability of EGISES and P-ROUGE. Through this research, we provide a more effective and comprehensive approach to evaluating personalized summarizer models, accounting for both, the accuracy and the personalized nature of the summaries.

List of Tables

| | | |
|-----|---|----|
| 5.1 | Test set format | 21 |
| 5.2 | Example of user written and model summary distribution (0.3805* value returned by OOVs Handling algorithm) | 26 |
| 5.3 | OOV handling in summary | 29 |
| 5.4 | Personalization vs. Accuracy w.r.t user profiles models | 32 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Change in the user profile and summary | 2 |
| 1.2 | Different evaluation metrics[3] | 2 |
| 1.3 | (a) shows accuracy is high but personalization is low. (b) shows both accuracy and personalization are high | 3 |
| 2.1 | Different types of text summarization models | 5 |
| 2.2 | Proposed human feedback, reward model training, and policy training procedure [13] | 8 |
| 2.3 | Human preference analysis [13] | 9 |
| 2.4 | Different evaluation metrics[3] | 11 |
| 2.5 | Pyramid evaluation model | 12 |
| 4.1 | Proportional difference of summary s_{i1} and user profile u_{i1} with rest of summaries and user profiles | 17 |
| 4.2 | Summary pair isn't deviating as per user profile pair | 19 |
| 5.1 | Test set creation process | 21 |
| 5.2 | Example of dataset | 21 |
| 5.3 | PENS framework [1] | 23 |
| 5.4 | Comparison of MLE loss and the contransitive loss [8] | 25 |
| 5.5 | Visualizing above distributions | 26 |
| 5.6 | RoBERTa follows BERT architecture [4] with some additional changes | 30 |
| 6.1 | ROUGE-L Dev and DINS Dev | 34 |
| 6.2 | Correlation with ROUGE-L Dev and DINS Dev using Pearson . . . | 34 |
| 6.3 | Correlation with ROUGE-L Dev and DINS Dev using Kendall . . . | 35 |
| 6.4 | Correlation with ROUGE-L Dev and DINS Dev using Spearman . . | 35 |
| 6.5 | ROUGE L vs Personalized ROUGE L | 37 |
| 6.6 | Correlation with ROUGE L and Personalized ROUGE L using Pearson | 37 |
| 6.7 | Correlation with ROUGE L and Personalized ROUGE L using Kendall | 38 |

| | | |
|-----|--|----|
| 6.8 | Correlation with ROUGE L and Personalized ROUGE L using Spearman | 38 |
| 7.1 | Direct agree meant of EGISES score by user | 39 |
| 7.2 | Annotator assign similarity score to summary pair | 40 |
| 7.3 | Illustration of score given by annotator | 40 |
| 7.4 | Online survey setup | 41 |

CHAPTER 1

Introduction

The growing availability of large-scale text data has led to an increasing demand for automated text summarization systems that aims to compress a lengthy document into a short paragraph, which includes salient information of that document. Since saliency is subjective, user might not be interested in the salient information presented in the summary. Most of the existing models generate generic summaries that may not be relevant or personalized to the user's specific interests. Hence, there should be a personalized summarization model that considers the user's attention/ interest while generating the summary[1].

Despite the importance of personalization in text summarization, there is still a lack of effective methods for measuring the degree to which the model is able to generate personalized summaries. This highlights the need for novel approaches to measure the degree of personalization of a summarization model.

1.1 Problem statement

The degree of personalization measures how well the model adapts to the user's preferences while generating a summary. In order to accurately measure it, proportional changes to both user profiles and summaries must be considered.

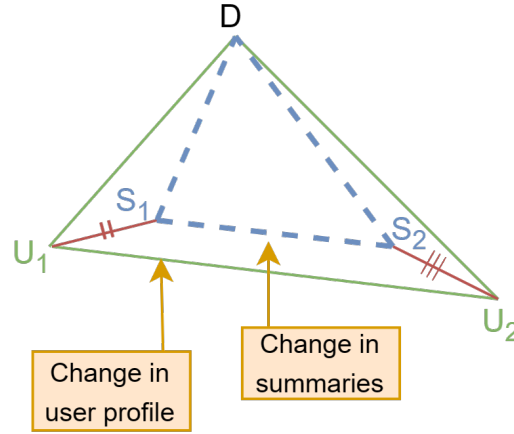


Figure 1.1: Change in the user profile and summary

1.2 Current work

There are several metrics that evaluate the model in terms of quality (focusing on the readability) and accuracy of the summary [3]. None of them evaluate how well a summarization model can capture user preference w.r.t. subjectivity (user’s individual perception of saliency). Measuring the degree of personalization is essential to know how well a summarization model can adapt to user preference while generating a personalized summary.

| | | Manual Evaluation | Automated Evaluation | |
|---------------------------|---------------------|---|---|---|
| | | | Reference-based | Reference-free |
| Accuracy | | 1. Factoid (2003) 2. Pyramid (2004) 3. SEE (2003) | 1. Cosine similarity (1989) 2. BLEU (2002) 3. ROUGE (2004) 4. Unit overlap (2002) 5. Latent-based (2009) 6. Semi-automated pyramid (2018) 7. Automated pyramid (2017) | 1. KL divergence (1959) 2. Jensen–Shannon divergence (1991) 3. FRESA (2013) 4. SummTriver (2018) 5. Summary likelihood (2013) 6. SUPERT (2020) |
| Quality | | 1. DUC 2005 readability 2. TAC 2008 readability | Sum-QE(2019) | |
| Degree of personalization | w.r.t. subjectivity | No explicit measure yet (as per our extensive study) | | |

Figure 1.2: Different evaluation metrics[3]

Exdos [5] based personalized summarization model [6] proposed by S. Ghodrattama was measured in terms of iterative convergence. Here convergence happens when the user stop giving feedback on the same document.

Microsoft proposed PENS framework[1], which can generate personalized head-

lines based on user profiles. Headlines can be considered as personalized summaries. To evaluate their model ROUGE was calculated between user-written personalized summary and model-generated personalized summary. We argue that ROUGE is an accuracy measure, it can not measure the degree of personalization. Fig. 1.3 (a) shows that the model has high accuracy since the user profile and summary have a small difference, indicating that the model captures users' interests quite well. However, personalization is low since their summary pair has less difference compared to the user profile pair. Fig. 1.3 (a) shows that the model has high accuracy while personalization is also high as the summary and the user profile pair have almost equal differences. So a model may have high accuracy but still have a low degree of personalization. The same is supported by our findings; more information on this will be provided in the results and discussion section. Thus, accuracy measures are not adequate for measuring the degree of personalization.

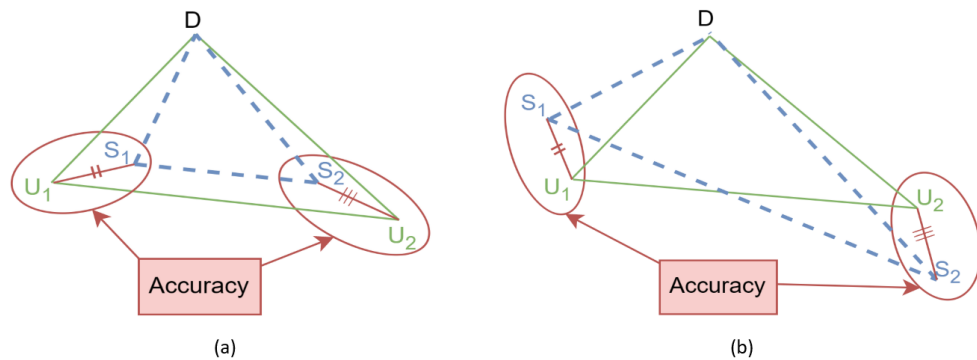


Figure 1.3: (a) shows accuracy is high but personalization is low. (b) shows both accuracy and personalization are high

The accuracy-centric measure focuses on the difference between user-written and model-generated summaries (shown by the red circle in fig. 1.3). In contrast, the personalization-centric measure focuses on the relationship between model-generated summaries and user profiles to evaluate the personalization capability of a summarization model.

1.3 Challenges

Post a thorough literature review, we found that there exist very few datasets in which multiple user-written summaries are available for a single document. Moreover, among these few datasets, non of them contain the user history, except the PENS dataset by Microsoft[1]. And this was the only dataset created with

the intention of developing a model that can generate personalized summaries. Thereby we found it the best suit for our research.

One of the biggest challenges was coming up with a formula that takes into account the user profile since it is the foundation for determining the degree of personalization and designing the formula required to discover and check various edge cases to ensure that the formula is functioning properly in all scenarios. During testing, we came across summary words that were not occurring in the document, such words are called OOV (out of vocabulary) words. so we developed an algorithm to handle these OOV words.

Due to the lack of personalization-centric measures, it is difficult to compare our metric with others. Therefore, we come up with direct and indirect ways to show reliability. We must gather survey data to demonstrate a correlation in order to establish direct human agreement, where the survey requires the involvement of human resources(annotators) and a server to deploy the survey application which will be further used by annotators.

1.4 Scope of research

The scope of this research is to evaluate the effectiveness of automated text summarization models in capturing subjectivity while generating summaries. While the focus of this research is on subjectivity, there are other important aspects of automated text summarization that can impact the user experience, such as temporal variance. This refers to how well and quickly a model can capture the user's drift of interest and adapt to changes in the user's preferences while generating summaries. Although not the primary focus of this research, these factors will also be taken into consideration during the evaluation of the models.

CHAPTER 2

Related work

In this section, we will discuss about related work in modeling and evaluation in the domain of text summarization.

2.1 Summarization Model

Based on the research so far, various types of text summarization can be categorized into the following categories, as depicted in the category diagram. The diagram illustrates different approaches and techniques used in text summarization and their examples, providing a visual representation of their classification.

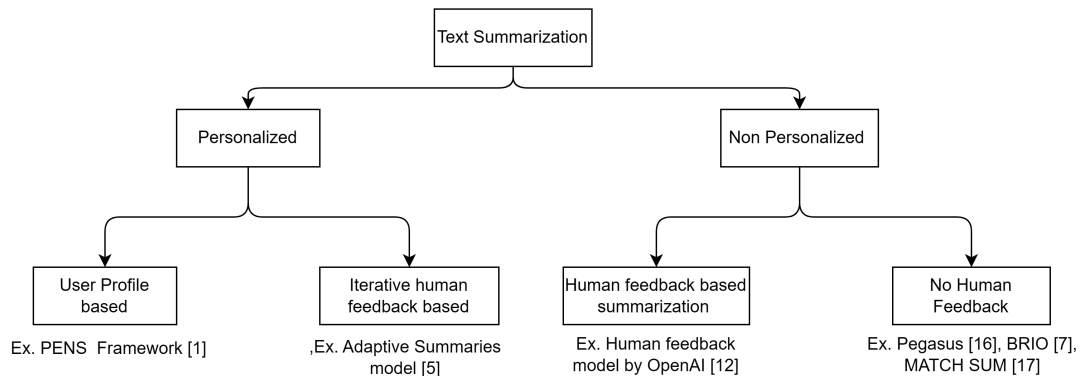


Figure 2.1: Different types of text summarization models

2.1.1 Personalized summarization models

Saliency is subjective hence depending on the user interest, the summary could differ essentially. Personalized summarization is a text summarization technique that considers and incorporates user preferences for the summarization task. These models aim to enhance user engagement and satisfaction by delivering more relevant summaries that align with the user's interests. By considering user-specific

factors, personalized summarization models can filter and prioritize information based on the user's preferences, resulting in summaries that are highly personalized and targeted [14].

Personalized summarization models can be further categorized into two categories: User Profile based and Interactive human feedback based

User Profile based personalization

In user profile based personalization models, user profiles can be created based on implicit or explicit feedback. Implicit feedback can be clicked article, reading time, scroll behaviour, and interaction with related content. Explicit feedback can be preference selection, ratings, annotation and highlights.

The PENS framework[1] proposes personalized news headlines generation using a Pointer-Generator Network as the base model. The training process involves training the network on actual headlines and using it to initialize the policy model. Reinforcement learning with Monte Carlo Tree Search is then applied to optimize the policy model. The reward for generating a headline is based on fluency, factual consistency, and coverage. The fluency reward is determined by a language model, while factual consistency and coverage are measured using ROUGE scores. Additionally, user embeddings are incorporated to generate personalized headlines by influencing the decoder's hidden state, attention weights, and word generation decisions. The experiment was performed on several user embeddings to check their impact on the performance of headline generation[1].

NAML[15] outperforms the remaining models used on the experiment on PENS framework. NAML stands for Neural news recommendation approach with attentive multi-view learning. The proposed model introduces an attentive multi-view learning approach to encode news representations from various perspectives, including titles, bodies, and topic categories. It utilizes both word-level and view-level attention networks to identify crucial words and views that contribute to informative news representations. Additionally, a user encoder is implemented to learn user representations based on their browsed news, taking into account the varying informativeness of different news articles. An attention mechanism is applied to the news encoder to select the most relevant and informative news for user representation learning. This comprehensive approach aims to capture the diverse aspects of news and user preferences, leading to enhanced news and user

representations within the model [15].

Iterative human feedback based personalization

In iterative human feedback based models, the user keeps giving feedback on the summary iteratively till the user satisfies with the model generated summary.

The Adaptive Summaries proposes a personalized concept-based summarization approach based on the idea of learning from users' feedback to generate summaries that are tailored to their individual needs and preferences. It uses exdos [5] as based extractive summarization model to rank sentences of news body. Further, system works by first extracting concepts from the input documents. Users are then presented with a list of concepts and asked to provide feedback on their importance. The system uses this feedback to generate a summary that includes the most important concepts. The summary is then presented to the user for further feedback. This process is repeated until the user is satisfied with the summary [6].

The Adaptive Summaries approach has several advantages over traditional summarization approaches. First, it is personalized to the individual user. This means that the summary is more likely to be relevant and interesting to the user. Second, the approach is interactive. This allows the user to provide feedback and influence the content of the summary[6].

2.1.2 Non Personalized summarization models

Conventional non-personalized text summarization refers to traditional approaches and techniques for generating summaries that are not tailored to individual users. These methods focus on extracting the most important information from a given text, such as articles, documents, or web pages, without considering user-specific factors. The goal of conventional non-personalized text summarization is to produce summaries that capture the essential points and main ideas of the source text, making them suitable for a broader audience.

Non personalized summarization models can be further categorized into two categories: Human feedback based summarization and No Human Feedback.

Human feedback based summarization

Human feedback model proposed by OpenAI [13] uses reward model trained by human feedback of choosing which summary is better. Results shows that a policy model trained on such reward model generates even better summary than human written summary. Procedure of training is as per following:

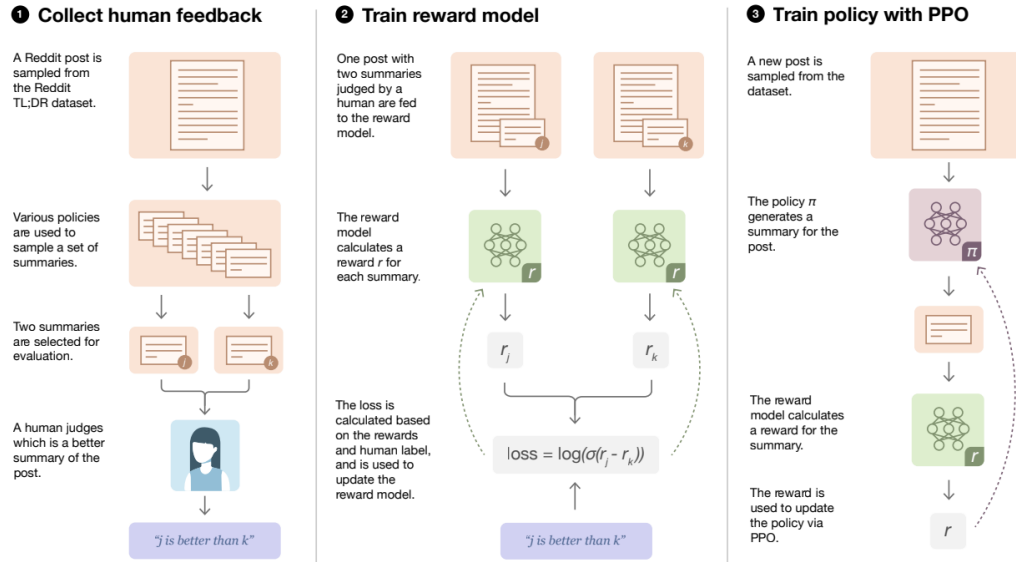


Figure 2.2: Proposed human feedback, reward model training, and policy training procedure [13]

- **Step 1: Collect Human Feedback**

In this initial step, human judges are asked to determine which summary is better between two options. These summaries are sampled from various sources, including the current policy, initial policy, original reference summaries, and different baselines. A detailed procedure is followed between the labelers and the researcher to ensure the quality of the labeling task. This involves the labelers reading the entire text first, forming their own interpretation, and only then assigning the label. Additionally, a certain level of agreement is required between the labelers and the researcher to ensure consistency and reliability in the labeling process [13].

- **Step 2: Train Reward Model**

This step involves training a reward model that can predict which summary is better among both. The reward model is trained based on the preference of the human judges. If a summary preferred by the human annotator, then the reward model also predicts that summary as a better summary [13].

- **Step 3: Train Policy with PPO**

In this step, a policy is trained using reinforcement learning (RL) techniques with the goal of generating high-quality summaries. The output of the reward model is treated as a reward for the entire summary, which is maximized using the Proximal Policy Optimization (PPO) algorithm. This means that the RL policy is updated to generate summaries that receive higher rewards from the reward model [13].

Fig. 2.3 shows that fraction of the time humans prefer proposed model’s summaries over the reference summaries of the Reddit dataset.

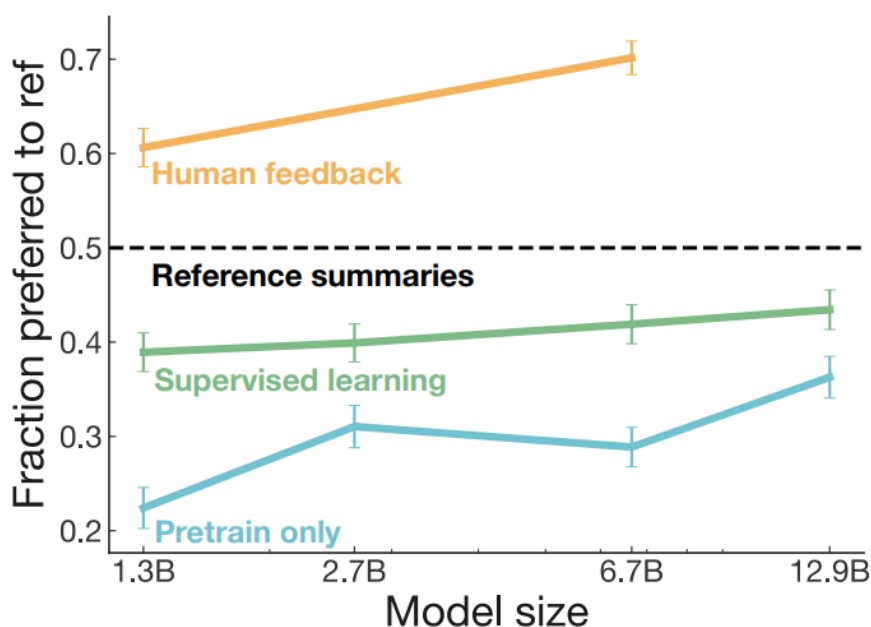


Figure 2.3: Human preference analysis [13]

No Human Feedback

Generic models like Pegasus [17] falls under this category, where only a document is given as input, and it will generate a generic summary. It aims to extract the most salient information from the overall document.

Pegasus aims to generate coherent and concise summaries from source documents. It utilizes a transformer-based architecture, specifically the encoder-decoder framework, to learn the relationships and semantic representations between the input text and the summary. Pegasus employs a pre-training and fine-tuning approach, where it is initially trained on a large corpus of publicly available text

from the internet and then fine-tuned on summarization-specific datasets. The model incorporates novel techniques such as a self-supervised objective called Gap Sentences Generation (GSG), which improves the quality and accuracy of the generated summaries. Pegasus has achieved state-of-the-art results on various summarization benchmarks and has demonstrated its effectiveness in generating abstractive summaries across a wide range of domains and languages [17].

The BRIO [8] paper introduces a novel training paradigm for abstractive summarization models, aiming to overcome performance degradation during inference. Existing models trained using maximum likelihood estimation suffer from exposure bias when comparing system-generated summaries with reference summaries. The proposed paradigm utilizes a non-deterministic distribution, assigning probabilities to candidate summaries based on their quality. This approach achieves state-of-the-art performance on CNN/DailyMail and XSum datasets, addressing exposure bias and improving overall model performance. The findings have implications beyond summarization, contributing to the advancement of natural language processing tasks [8].

The paper [18] introduces a novel approach to neural extractive summarization systems, presenting the task as a semantic text matching problem. The authors emphasize the need for this paradigm shift, highlighting the inherent gap between sentence-level and summary-level extraction. To substantiate their claim, they conduct a comprehensive analysis of this gap, thoroughly examining the properties of the dataset. In response to this, the paper introduces an innovative summary-level framework named MATCH SUM. This framework reconceptualizes extractive summarization by considering it as a semantic text matching challenge. The underlying idea is that a high-quality summary should exhibit greater semantic similarity to the source document as a whole, distinguishing it from less qualified summaries. Through this proposed approach, the paper aims to enhance the effectiveness and coherence of extractive summarization systems [18].

2.2 Summarization Evaluation

There are several evaluation metrics available focusing on different aspects of summary, such as accuracy and quality. Evaluation can be done manually or automatic. In case of manual evaluation human need to annotate data manually,

which require high cost. In case of automatic evaluation, where model performance evaluated by system, no human interaction required. When reference summary available, automatic metric evaluate based on overlapping between reference and model generate summary. When reference summaries are not available, reference free metrics evaluate performance based by checking similarity between model generated summary and document.

| | Manual Evaluation | Automated Evaluation | |
|----------|---|---|---|
| | | Reference-based | Reference-free |
| Accuracy | <ol style="list-style-type: none"> 1. Factoid (2003) 2. Pyramid (2004) 3. SEE (2003) | <ol style="list-style-type: none"> 1. Cosine similarity (1989) 2. BLEU (2002) 3. ROUGE (2004) 4. Unit overlap (2002) 5. Latent-based (2009) 6. Semi-automated pyramid (2018) 7. Automated pyramid (2017) | <ol style="list-style-type: none"> 1. KL divergence (1959) 2. Jensen–Shannon divergence (1991) 3. FRESA (2013) 4. SummTriver (2018) 5. Summary likelihood (2013) 6. SUPERT (2020) |
| Quality | <ol style="list-style-type: none"> 1. DUC 2005 readability 2. TAC 2008 readability | Sum-QE(2019) | |

Figure 2.4: Different evaluation metrics[3]

2.2.1 Accuracy

In manual accuracy evaluation, Pyramid method [11] is one of widely used approach. It works as follows : From cluster of documents of a topic, four annotator write own summary and then another annotator will read all these summaries and generate SCUs (Summary Content Units), which are labels or keywords. Evaluator will then use these SCUs and manually assigns score based to respective summary if that SCUs contain in that summary [11].

ROUGE and BLEU are widely accepted automatic reference based measures that assigns score to summary based on N gram overlapping between reference and model generated summary.

SummTriver[2] is reference free metric that uses probability distribution to assign score to the summary. Let there are multiple summary available for a document now task is to know which summary is best based on score given by SummTriver, then first we need R, P and Q where R is distribution generated by the summary to evaluate, P is distribution obtained from a collection of summaries (different from R) of same document and Q represents probability distribution of source document. Every summary from P can be analyzed with SummTriver, by interchanging summary in R by one from P. Probability distributions' events represented as single type of word n-grams: unigrams, bigrams or skip-bigrams. SummTriver

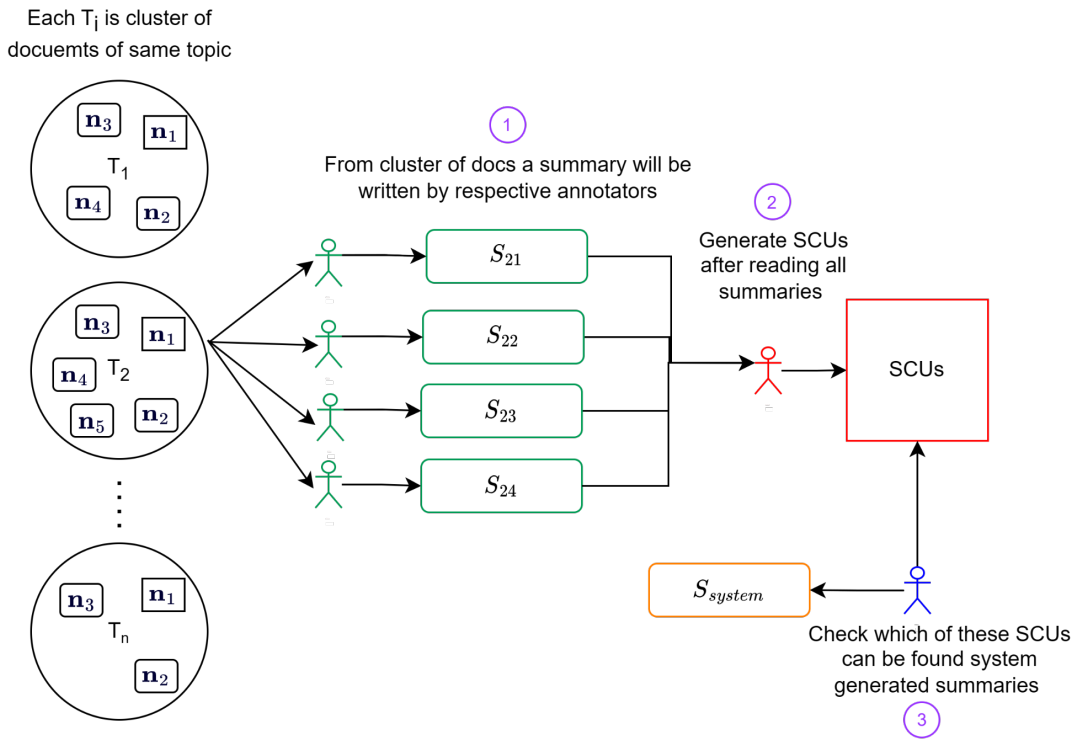


Figure 2.5: Pyramid evaluation model

compare how different is summary R and set of summaries P with respect to the source document Q [2].

2.2.2 Quality

Quality-based approach, DUC 2005 readability was partially employed in evaluating the summaries presented by participants in the DUC 2005 competition. It assesses summary quality based on several dimensions, including grammatical correctness, non-redundancy, referential clarity, focus, structure, and coherence. These dimensions ensure that summaries are free from grammatical errors, avoid unnecessary repetition, maintain clear references, contain relevant information, and exhibit well-structured organization. Human experts evaluate the summaries using a scale ranging from very poor to very good. TAC 2008 also adopts this quality-based approach, assessing summaries based on the same dimensions and scale as DUC 2005 readability [3].

CHAPTER 3

Degree of personalization

In the field of personalized text summarization, the degree of personalization refers to the level of personalization that a model can provide to meet the user's preferences and interests while generating a summary. The degree of personalization can be measured by evaluating the model's ability to adapt to changes in the user profile, including changes in the user's interests and preferences, and generating summaries that reflect these changes. This ensures that the summaries generated by the model are according to the user's specific needs and are not generic. The degree of personalization is a critical factor in developing and evaluating personalized text summarization models as it directly impacts the user experience. A high degree of personalization indicates that the model-generated summaries are more relevant, engaging, and useful to the user, while a low degree of personalization may result in less satisfactory summaries. Therefore, it is essential to consider the degree of personalization while developing and evaluating automated text summarization systems to provide a better user experience.

One way to look at the degree of personalization is how sensitive or insensitive the model is in capturing user interest. If model is highly insensitive in capturing user interest, then degree of personalization will be low as the model-generated summaries will not accurately reflect the user's preferences. As a result, the summaries may not be relevant, engaging, or useful to the user, leading to a lower degree of personalization.

3.1 Defining Insensitivity to subjectivity

Insensitivity to subjectivity refers to a situation when a model fails to capture an individual's preferences, perspectives, or interests when generating a summary. Suppose the model generates similar summaries for two different user profiles. In that case, it suggests that the model does not consider the differences in the

users' interests, and thus it is insensitive to subjectivity. This insensitivity results in badly personalized summaries, which may not meet individual users' unique needs and preferences.

Mathematically, given document \mathbf{D} , and user profile details \mathbf{u} , let a summarization model $M_{\theta, \mathbf{u}}$ generate the best estimated personalized summary \hat{S}_u .

$$M_{\theta, \mathbf{u}} : \mathbf{D}, \mathbf{u} \mapsto \hat{S}_u$$

Further, given two different user profiles, \mathbf{u}_i and \mathbf{u}_j , summarization model $M_{\theta, \mathbf{u}}$ is (weakly) *Insensitive-to-Subjectivity* iff $\forall(\mathbf{u}_i, \mathbf{u}_j), f_{dist}^U(\mathbf{u}_i, \mathbf{u}_j)^* > \tau_{max}^U$:

$$f_{sim}^S(M_{\theta, \mathbf{u}}(\mathbf{D}, \mathbf{u}_i), M_{\theta, \mathbf{u}}(\mathbf{D}, \mathbf{u}_j)) < \tau_{min}^S$$

where

- f_{dist}^U : User profile distance function
- f_{sim}^S : Summary similarity function
- τ_{max}^U : Max. limit for two different user profiles to be mutually indistinguishable
- τ_{min}^S : Min. limit for two generated summary w.r.t two different users to be mutually distinguishable

$$* : f_{dist}^U(\mathbf{u}_i, \mathbf{u}_i) = 0 \text{ and } f_{dist}^U(\mathbf{u}_i, \mathbf{u}_j) \in [0, 1]$$

Consider table 3.1 as example of how we can compare Degree-of-Insensitivity w.r.t subjectivity of different summarization models (let say denoted by $\sigma_{sub}(D, M_{\theta_x, \mathbf{u}})$, $M_{\theta_y, \mathbf{u}}$ and $M_{\theta_z, \mathbf{u}}$. If we have a metric that gives a score based on how insensitive the model is, i.e., how poorly a model generates a personalized summary, then as per table 3.1, a high score in $\sigma_{sub}(D, M_{\theta_x, \mathbf{u}})$ indicates poor personalized summary score given by metric, while a low score in $\sigma_{sub}(D, M_{\theta_x, \mathbf{u}})$ indicates model generated better personalized summary for that news article. The expected model score $\sigma_{sub}(M_{\theta_x, \mathbf{u}})$ is sample-average of its $\sigma_{sub}(D, M_{\theta_x, \mathbf{u}})$ score of all news over all user pairs.

| \mathbf{u}_i | \mathbf{u}_j | Document | $\sigma_{sub}(D, M_{\theta_x, \mathbf{u}})$ | $\sigma_{sub}(D, M_{\theta_y, \mathbf{u}})$ | $\sigma_{sub}(D, M_{\theta_z, \mathbf{u}})$ |
|----------------|----------------|-------------------------|---|---|---|
| Bob | Alice | News₁ | 0.43 | 0.27 | 0.61 |
| | | News₂ | 0.32 | 0.86 | 0.52 |
| | | News₃ | 0.58 | 0.51 | 0.39 |

Expected model score $\sigma_{sub}(M_{\theta_x, \mathbf{u}})$ is sample-average of its $\sigma_{sub}(D, M_{\theta_x, \mathbf{u}})$ score of all news over all user pairs

CHAPTER 4

Measuring Degree of Personalization

We will talk about our proposed measure, EGISES, in this chapter. It makes use of a deviation function to calculate the deviation of a summary with rest of user-profiles and summaries of that document.

4.1 Deviation

Deviation calculates the proportional difference between a summary \mathbf{s}_{ij} and its corresponding user profile \mathbf{u}_{ij} , with respect to other model-generated summaries and user profiles of the same document. Specifically, $\text{Dev}(\mathbf{s}_{ij} | \mathbf{u}_{ij})$ that is the deviation of \mathbf{s}_{ij} given \mathbf{u}_{ij} measures the deviation of the j^{th} summary of the i^{th} document from the rest of k summaries of the same document and likewise the corresponding j^{th} user profile's deviation from rest of user profiles of the same document. As show in figure 4.1, deviation is calculated by comparing the summary \mathbf{s}_{i1} and user profile \mathbf{u}_{i1} with all other user profiles ($\mathbf{u}_{i2}, \mathbf{u}_{i3}, \dots, \mathbf{u}_{i5}$) and summaries ($\mathbf{s}_{i2}, \mathbf{s}_{i3}, \dots, \mathbf{s}_{i5}$) generated by the model, respectively. Here, we used JSD (Jensen–Shannon divergence) to find the distance between the user profile pair and the summary pair, as JSD is symmetrized version of the Kullback–Leibler divergence, which measures the similarity between two distributions [10]. Further, it is weighted by how far it deviates from the actual document using $w(\mathbf{u}_{ij}, \mathbf{u}_{ik})$ which is the ratio of the distance between two users and the distance between user and document. Softmax is applied on these weights so that weights remain in the range of 0 and 1, the same goes for $w(\mathbf{s}_{ij}, \mathbf{s}_{ik})$ as well.

A lower value of deviation indicates that user profiles are different but the summaries are very similar to other model-generated summaries or vice versa, hence are not well personalized. On the other hand, a higher value of deviation indicates that the user profiles are different at the same time summary are also that much different from other summaries and hence are well personalized.

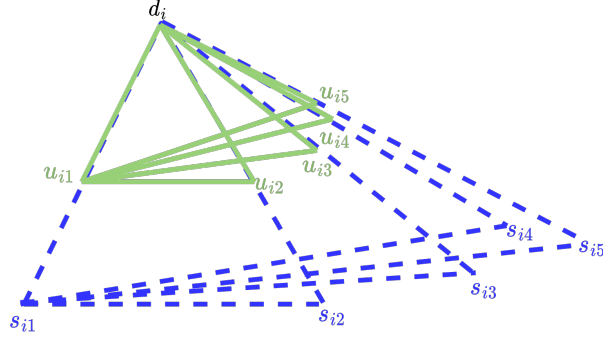


Figure 4.1: Proportional difference of summary s_{i1} and user profile u_{i1} with rest of summaries and user profiles

$$Dev(\mathbf{s}_{ij}|\mathbf{u}_{ij}) = \frac{1}{n} \sum_{k=1}^n \frac{\min(u_score_{ijk}, s_score_{ijk})}{\max(u_score_{ijk}, s_score_{ijk})}$$

$$u_score_{ijk} = \frac{\exp(w(\mathbf{u}_{ij}, \mathbf{u}_{ik}))}{\sum_{l=1}^n \exp(w(\mathbf{u}_{ij}, \mathbf{u}_{il}))} * JSD(\mathbf{u}_{ij}||\mathbf{u}_{ik})$$

$$s_score_{ijk} = \frac{\exp(w(\mathbf{s}_{ij}, \mathbf{s}_{ik}))}{\sum_{l=1}^n \exp(w(\mathbf{s}_{ij}, \mathbf{s}_{il}))} * JSD(\mathbf{s}_{ij}||\mathbf{s}_{ik})$$

$$w(\mathbf{u}_{ij}, \mathbf{u}_{ik}) = \frac{JSD(\mathbf{u}_{ij}||\mathbf{u}_{ik})}{JSD(\mathbf{u}_{ij}||\mathbf{d}_i)}$$

$$w(\mathbf{s}_{ij}, \mathbf{s}_{ik}) = \frac{JSD(\mathbf{s}_{ij}||\mathbf{s}_{ik})}{JSD(JSD(\mathbf{s}_{ij}||\mathbf{d}_i))}$$

$$w(\mathbf{u}_{ij}, \mathbf{u}_{il}) = \frac{JSD(\mathbf{u}_{ij}||\mathbf{u}_{il})}{JSD(\mathbf{u}_{ij}||\mathbf{d}_i)}$$

$$w(\mathbf{s}_{ij}, \mathbf{s}_{il}) = \frac{JSD(\mathbf{s}_{ij}||\mathbf{s}_{il})}{JSD(JSD(\mathbf{s}_{ij}||\mathbf{d}_i))}$$

$$JSD(\mathbf{u}_{ij}||\mathbf{u}_{ik}) = \frac{1}{2} \cdot D_{KL}(\mathbf{u}_{ij}||\mathbf{m}) + \frac{1}{2} \cdot D_{KL}(\mathbf{u}_{ik}||\mathbf{m})$$

$$\mathbf{m} = \frac{1}{2} \cdot (\mathbf{u}_{ij} + \mathbf{u}_{ik})$$

$$D_{KL}(\mathbf{u}_{ij}||\mathbf{m}) = \sum_{p=1}^V \mathbf{u}_{ijp} \log \left(\frac{\mathbf{u}_{ijp}}{\mathbf{m}_p} \right)$$

$$D_{KL}(\mathbf{u}_{ik}||\mathbf{m}) = \sum_{p=1}^V \mathbf{u}_{ikp} \log \left(\frac{\mathbf{u}_{ikp}}{\mathbf{m}_p} \right)$$

$$JSD(\mathbf{s}_{ij}||\mathbf{s}_{ik}) = \frac{1}{2} \cdot D_{KL}(\mathbf{s}_{ij}||\mathbf{n}) + \frac{1}{2} \cdot D_{KL}(\mathbf{s}_{ik}||\mathbf{n})$$

$$\mathbf{n} = \frac{1}{2} \cdot (\mathbf{s}_{ij} + \mathbf{s}_{ik})$$

$$D_{KL}(\mathbf{s}_{ij}||\mathbf{n}) = \sum_{p=1}^V \mathbf{s}_{ijp} \log \left(\frac{\mathbf{s}_{ijp}}{\mathbf{n}_p} \right)$$

$$D_{KL}(\mathbf{s}_{ik}||\mathbf{n}) = \sum_{p=1}^v \mathbf{s}_{ikp} \log \left(\frac{\mathbf{s}_{ikp}}{\mathbf{n}_p} \right)$$

where

- v : vocabulary size
- n : number of personalized summaries for a single document
- \mathbf{d}_i : i^{th} document representation (distribution over the vocabulary v)
- \mathbf{s}_{ij} : j^{th} model generated summary representation (distribution over the vocabulary v) of i^{th} document
- \mathbf{u}_{ij} : j^{th} user profile representation (distribution over the vocabulary V) of i^{th} document
- u_score_{ijk} : calculates weighted divergence between j^{th} and k^{th} user pairs of i^{th} document
- s_score_{ijk} : calculates weighted divergence between j^{th} and k^{th} summary pairs of i^{th} document
- $JSD(\mathbf{u}_{ij}||\mathbf{u}_{ik})$: calculates Jensen Shannon divergence between distribution \mathbf{u}_{ij} & \mathbf{u}_{ik}
- $D_{KL}(\mathbf{u}_{ij}||\mathbf{m})$: calculates Kullback-Leibler Divergence between distribution \mathbf{u}_{ij} & \mathbf{m}
- $w(\mathbf{u}_{ij}, \mathbf{u}_{ik})$: calculates ratio of distance between user profile pair (\mathbf{u}_{ij} & \mathbf{u}_{ik}) and distance between user profile and document (\mathbf{u}_{ij} & \mathbf{d}_i)

4.2 Effective Degree of Insensitivity w.r.t subjectivity (EGISES)

To determine the degree of insensitivity of personalized summarization models, we have proposed a novel measure called EGISES (Effective Degree of Insensitivity w.r.t subjectivity). This measure evaluates the performance of the model based on how much the summary deviates from the user profile.

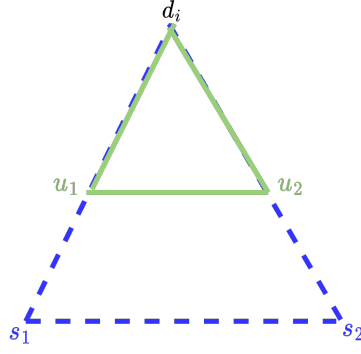


Figure 4.2: Summary pair isn't deviating as per user profile pair

A high value of EGISES indicates that the summary pair does not deviate as much as the user profile pair (Figure 4.2). This implies that the model is not that capable of capturing personalization effectively and is thus insensitive to the user's preferences and needs. As a result, a model with a high EGISES score would be considered bad at generating personalized summaries. Therefore, EGISES can be a valuable measure in assessing the performance of such models and can help in the development of more effective personalized summarization systems.

$$EGISES = 1 - \frac{1}{m*n} \sum_{\mathbf{d}_i:i=1}^m \sum_{(\mathbf{s}_{ij}, \mathbf{u}_{ij}):j=1}^n Dev(\mathbf{s}_{ij}|\mathbf{u}_{ij})$$

where

- m : Number of documents
- n : Number of personalized summaries in a document
- \mathbf{s}_{ij} : j^{th} model generated summary of i^{th} document
- \mathbf{u}_{ij} : j^{th} user profile of i^{th} document
- \mathbf{d}_i : i^{th} document
- $Dev(\mathbf{s}_{ij}|\mathbf{u}_{ij})$: deviation of summary \mathbf{s}_{ij} given user profile \mathbf{u}_{ij}

EGISES $\in [0, 1]$

CHAPTER 5

Case study on PENS dataset and framework

Post a thorough literature review, we find that there exist a very few datasets in which multiple user-written summaries are available for a single document. Moreover, among these few datasets, non of them contain the user history, except the PENS dataset by Microsoft[1]. And this was the only dataset created with the intention of developing a model that can generate personalised summaries. Thereby we found it the best suit for our research.

5.1 PENS Dataset

Test data from the PENS dataset[1] was used in our study, and it consisted of headlines that could be considered as TLDR summaries of news articles. Test set was created in two phases. In the first phase, they collected data from 103 native English speakers who were asked to browse through 1,000 news headlines and mark at least 50 pieces they were interested in. The headlines were randomly selected and arranged according to their first exposure time.

The second stage of the test involved asking the participants to write their preferred headlines for 200 different articles, without knowing the original news title. These news articles were excluded from the first stage and were redundantly assigned to ensure that each news article was seen by an average of 4 people. The participants' click behaviours and more than 20,000 manually-written personalized headlines of news articles were also collected, which were regarded as the gold standard of user-preferred titles[1].

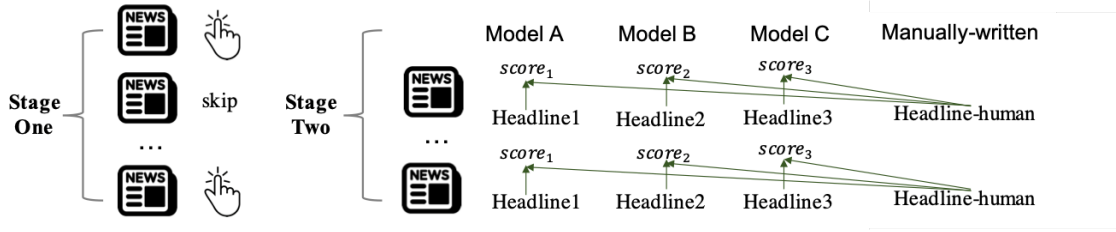


Figure 5.1: Test set creation process

Attributes of the test set are as per mentioned in table 5.1.

| Column | Example | Description |
|----------------|------------------------|---|
| userid | NT1 | The unique ID of 103 users |
| clicknewsID | N108480,N38238, ... | User's historical clicked news at 1st stage |
| posnewID | N24110,N62769, ... | Exhibited news for each user at 2nd stage |
| rewrite_titles | 'Legal battle looms... | Manuallywritten news for exhibited news |

Table 5.1: Test set format

In fig. 5.2, underlined words and colored words represent the correlated words in the manually-written headlines, clicked news, and generated headlines, respectively.

| | |
|-----------------------------------|--|
| Case 1. Original Headline: | Venezuelans rush to Peru before new requirements take effect |
| Pointer-Gen: | Venezuelans rush to Peru |
| user A written headline: | New <u>requirements</u> set to take effect causes <u>Venezuelans</u> to rush to <u>Peru</u> |
| NAML+HG for user A: | <u>Peru</u> has stricter entry <u>requirements</u> for escaping <u>Venezuelans</u> on that influx. |
| Clicked News of user A: | 1. <u>Peru</u> and <u>Venezuela</u> fans react after match ends in a draw 2. <u>Uruguay v. Peru</u> , <u>Copa America</u> and <u>Gold Cup</u> , <u>Game threads</u> and <u>how to watch</u> |
| user B written headline: | <u>Venezuelan migrants</u> to Peru face danger and discrimination |
| NAML+HG for user B: | Stricter entry requirements on <u>Venezuelan migrants</u> and <u>refugees</u> . |
| Clicked News of user B: | 1. Countries Accepting The Most <u>Refugees</u> (And Where They're Coming From) 2. <u>Venezuelan</u> mothers, children in tow, rush to <u>migrate</u> |

Figure 5.2: Example of dataset

5.2 PENS framework

The paper proposes a novel framework for generating personalized news headlines that takes into account both the content of the news article and the user's reading interests. Given a user u and its past click history $[c_1^u, c_2^u, \dots, c_N^u]$ here, each variable c represents the headline of a news article that a user u has clicked on. The headline is composed of a sequence of words, denoted as $c = [w_{c_1}, w_{c_2}, \dots, w_{c_T}]$, where T is the maximum length of the headline. Given the news body of another

article $c = [w_{v_1}, w_{v_2}, \dots, w_{v_n}]$ that is being presented to the same user u , the objective is to generate a personalized news headline $H_v^u = [y_{v_1}^u, y_{v_2}^u, \dots, y_{v_T}^u]$ based on the previously clicked news articles $[c_1^u, c_2^u, \dots, c_N^u]$ and the news body v of current news article [1].

The authors use a Pointer-Generator Network as base model. In the training process, a Pointer-Generator Network trained on actual headlines. Further this model used to initialize the policy model, which is then optimized using reinforcement learning with a Monte Carlo Tree Search algorithm. The reward for generating a headline is calculated based on three factors: fluency, factual consistency, and coverage. Fluency is assessed by a language model that uses a two-layer LSTM pre-trained on news body data. The probability estimation of a generated headline is considered as the fluency reward. Factual consistency and coverage are measured by calculating the mean of ROUGE-1, ROUGE-2, and ROUGE-L F-scores between each sentence in the news body and the generated headline. The top three scores are averaged to obtain the reward. All three rewards are then averaged to produce a final signal. As these reward functions only produce an end reward after the whole headline is generated, a Monte Carlo Tree search is applied to estimate intermediate rewards [1].

User embedding is also given as an input along with the news article to generate personalized headlines. Experiments were done on different techniques like NAML [15], ENBR [12], NRMS [16] to generate user embeddings. The paper introduces three approaches for incorporating user embeddings to generate personalized news headlines. Firstly, the user embedding is utilized to initialize the decoder's hidden state, enabling the model to consider the user's reading preferences from the beginning of the headline generation process. Secondly, personalized attentive values are employed to assign higher attention weights to words in the news body that are more relevant to the user's interests. This ensures that those words receive greater focus during headline generation. Lastly, the user embedding influences the decision between word generation and copying. If a word in the news body matches a word in the user's interest vocabulary, it is more likely to be directly copied into the generated headline rather than being created anew. These three methods collectively enhance the modeling of individual preferences and interests, resulting in personalized headlines of higher quality[1].

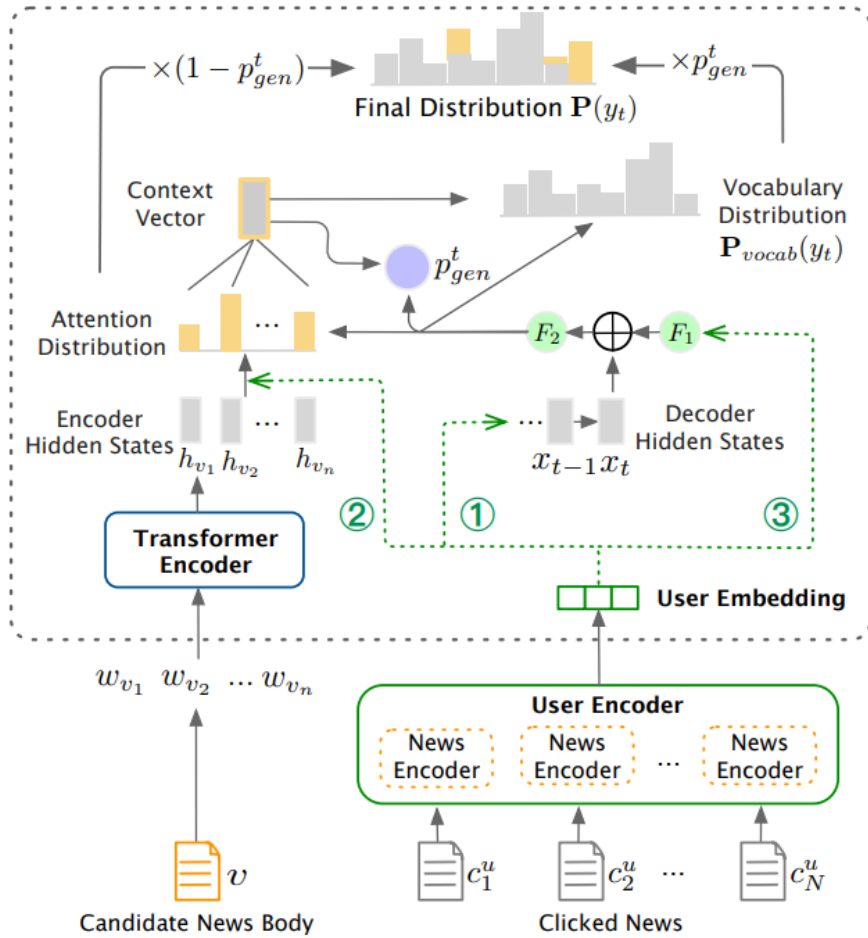


Figure 5.3: PENS framework [1]

5.3 BRIO (Bringing Order to Abstractive Summarization)

Other than PENS framework which is a personalized summarization model, we experiment with a non-personalized model as well to see how it performs with a slightly modified setup. Since the non-personalized model just takes the document and gives a summary as output without taking into consideration of user interest to provide user information, i.e., what the user is expecting in summary based on its interest, we concatenate the personalized summary written by that user in PENS dataset with the news body of that document. Now this modified news body is given as input to any non-personalized model. For our experiment, we used BRIO model [8].

The BRIO paper proposes a new training paradigm for abstractive summarization models that address the problem of performance degradation during inference. Abstractive summarization models are commonly trained using maximum likelihood estimation, which assumes a deterministic target distribution where an ideal model assigns all probability mass to the reference summary. However, during inference, the model needs to compare several system-generated candidate summaries that may deviate from the reference summary, leading to exposure bias and decreased performance. To address this problem, the proposed training paradigm assumes a non-deterministic distribution where different candidate summaries are assigned probability mass according to their quality [8].

As shown in fig.5.4, MLE takes into account a deterministic distribution with an all probability mass assigned to the reference summary. The proposed method, in contrast, makes the non-deterministic assumption that the quality of system-generated summaries also affects their probability mass. The contrastive loss aligns the model's predicted probabilities of candidate summaries with the actual quality metric M used for evaluation. Their proposed approach allows the abstractive model to serve both as a generation model and a reference-free evaluation model simultaneously [8].

This method achieves state-of-the-art results on the CNN/DailyMail and XSum datasets and can estimate probabilities of candidate summaries that are more correlated with their level of quality. The paper highlights the importance of addressing exposure bias in abstractive summarization models and proposes a novel solution that improves performance during inference. The results demonstrate the effectiveness of this approach and suggest its potential for improving other natu-

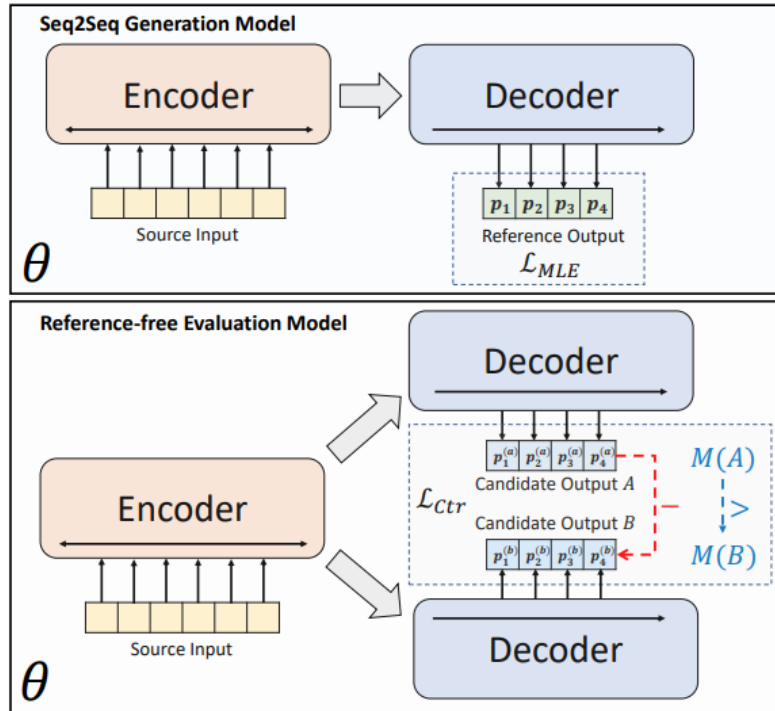


Figure 5.4: Comparison of MLE loss and the contrasitive loss [8]

ral language processing tasks like summarization [8].

5.4 Experimental setup to compute EGISES on PENS dataset

There are several ways to check how a summary pair deviates from each other. In the case of vector space, first embedding is generated for both summaries and then using similarity measure, the distance between summaries can be calculated. While in the case of probability distribution space, by calculating divergence, one can measure how one summary deviates from the other. If both distributions are very close or have high overlap, it indicates the summary pair is highly similar i.e., very close to each other in terms of distance. In our research, we followed probability distribution space to find deviation. Hence first, we need to calculate the distribution for Document D , Summary S , and User profile U .

To calculate the distribution of model summaries, we can analyze the word count of the generated summaries. On the other hand, we can use the distribution of user-written personalized summaries as the user profile. Because a user writes their own summary based on his subjectivity of saliency. Hence, the user writes

keywords/topics that he/she is most interested in with regard to the article.

Consider below example:

Document: red cat on red tall table -> Preprocessing -> [red, cat, red, tall, table]

User-written Summary: cat on table -> Preprocessing -> [cat, table]

Model Summary: red cat on desk (OOV) -> Preprocessing -> [red, cat, desk]

| Vocab | Doc distribution | User summary distribution | Model summary distribution |
|-------|------------------|---|--|
| red | $2/5 = 0.4$ | 0 (absent) | $(1/3) / (1/5) = 1.66 \Rightarrow 1.66/3.7005 = 0.4485$ |
| cat | $1/5 = 0.2$ | $(1/2) / (1/5) = 2.5 \Rightarrow 2.5/5 = 0.5$ | $(1/3) / (1/5) = 1.66 \Rightarrow 1.66/3.7005 = 0.4485$ |
| tall | $1/5 = 0.2$ | 0 (absent) | 0 (absent) |
| table | $1/5 = 0.2$ | $(1/2) / (1/5) = 2.5 \Rightarrow 2.5/5 = 0.5$ | 0 (absent) |
| desk | 0 (absent) | 0 (absent) | $(1/3) / ? = ?$ (OOV) $\Rightarrow 0.3805^*/3.7005 = 0.1028$ |

Table 5.2: Example of user written and model summary distribution (0.3805* value returned by OOVs Handling algorithm)

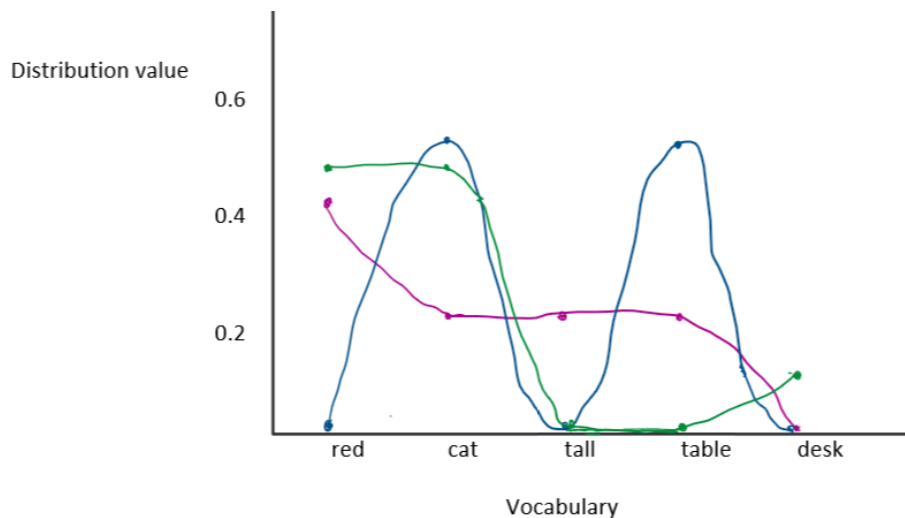


Figure 5.5: Visualizing above distributions

The processing performed on the document, model summary, and user-written summary to tokenized, remove stop words and lemmatization. Document distribution is generated by taking the ratio of word count in the document and the total number of words in the document for each word in the document, that is *word count in document / total number of words in document*. Likewise, user summary distribution is generated by taking the ratio of ratio of words in the user summary and the ratio of words in the document. And Model summary distribution is generated by taking the ratio of words in the model summary and the ratio of words in the document, for each word in the model summary. In above

model distribution have OOV word, we got score 0.3805 by applying OOVs handling algorithm, which you can find in next section, Algorithm for is Distribution Generation as following.

Algorithm 1 Distribution Generation of a document and its respective user and model summary

- 1: Apply preprocessing on Document, Model Summary, and User Summary
 - 2: Create vocabulary set by considering all words of preprocessed Document, Model Summary, and User Summary $total_words$
 - 3: Initialize distribution of Document, Model Summary, and User Summary with vocabulary set
 - 4: **for** each word w in Document **do**
 - 5: Calculate word count in Document: $count \leftarrow$ count of occurrences of w in Document
 - 6: Calculate ratio of word count in Document: $ratio_doc \leftarrow \frac{count}{total_words_in_doc}$
 - 7: Assign $ratio_doc$ as distribution value for word w in Document Distribution
 - 8: **for** each word w in User Summary **do**
 - 9: **if** w is OOV **then**
 - 10: Assign the value returned by OOV Handling algorithm
 - 11: **else**
 - 12: Calculate ratio of words in User Summary:
 - 13: $ratio_u_summ \leftarrow \frac{occurrences_of_w_in_user_summ}{total_words_in_user_summ}$
 - 14: Calculate ratio of word count in Document:
 - 15: $ratio_doc \leftarrow \frac{occurrences_of_w_in_doc}{total_words_in_doc}$
 - 16: Calculate ratio of words in User Summary: $user_ratio \leftarrow \frac{ratio_u_summ}{ratio_doc}$
 - 17: Assign $user_ratio$ as distribution value for word w in User Summary Distribution
 - 18: **for** each word w in Model Summary **do**
 - 19: **if** w is OOV **then**
 - 20: Assign the value returned by OOV Handling algorithm
 - 21: **else**
 - 22: Calculate ratio of words in Model Summary:
 - 23: $ratio_m_summ \leftarrow \frac{occurrences_of_w_in_model_summ}{total_words_in_model_summ}$
 - 24: Calculate ratio of word count in Document:
 - 25: $ratio_doc \leftarrow \frac{occurrences_of_w_in_doc}{total_words_in_doc}$
 - 26: Calculate ratio of words in Model Summary: $model_ratio \leftarrow \frac{ratio_m_summ}{ratio_doc}$
 - 27: Assign $model_ratio$ as distribution value for word w in Model Summary Distribution
 - 28: $u_sum \leftarrow$ sum of all element of User Summary Distribution
 - 29: **for** each word w in User Summary Distribution **do**
 - 30: $value_of_w \leftarrow \frac{value_of_w}{u_sum}$
 - 31: $m_sum \leftarrow$ sum of all element of Model Summary Distribution
 - 32: **for** each word w in Model Summary Distribution **do**
 - 33: $value_of_w \leftarrow \frac{value_of_w}{m_sum}$
 - 34: **Return** Document Distribution, Model Summary Distribution, User Summary Distribution
-

5.4.1 OOV Handling

Words that occur in summary but not the original text are known as out-of-Vocabulary (OOV) words. We developed the following algorithm to handle OOVs.

Algorithm 2 OOVs Handling

```
1: for  $words\_in\_doc = w_1, w_2, \dots, N$  do
2:   Get embedding vector for oov and  $w_i$  (we used RoBERTa[9])
3:   Find cosine similarity between embeddings of oov and  $w_i$ 
4:  $max\_sim\_score =$  maximum similarity score among all words in doc
5:  $bias = 1 - \sqrt{max\_sim\_score}$ 
6: if ( $bias > max\_sim\_score$ )
7:   Return 0
8: else
9:   Apply softmax over all similarity scores to convert into a probability score
10: Return (ratio of oov in model summary) / (sum of all probability score)
```

Continuation of previous example 5.3, now we can generate value for OOV words. In this case, bias not have maximum similarity score in which indicates its not OOV word there might be word in document that have same meaning as OOV word, hence we pass summation of softmax score of all words. So distribution value of desk in model summary will be $((1/3)/0.876) = 0.3805$

| Words in doc | Similarity of each word in doc with "desk" | Softmax |
|--------------|--|---------|
| red | 0.537 | 0.2 |
| cat | 0.405 | 0.175 |
| tall | 0.613 | 0.216 |
| table | 0.892 | 0.285 |
| bias | $1 - (0.89)1/2 = 0.056$ | 0.124 |

Table 5.3: OOV handling in summary

RoBERTa [9], an enhancement over BERT [4] (Bidirectional Encoder Representations from Transformers), shares the same underlying architecture as shown in fig. but incorporates notable modifications to achieve better performance. RoBERTa employs a larger training corpus, including BooksCorpus, English Wikipedia, Common Crawl, and CC-News datasets, surpassing BERT’s training data. Moreover, RoBERTa extends the training duration and adopts a longer sequence length. BERT was trained for 1 million steps with a sequence length of 128, while RoBERTa is trained for 160,000 steps with a sequence length of 512. In terms of masking strategy, RoBERTa focuses solely on masked language modeling (MLM), eliminating the next sentence prediction (NSP) task. Additionally, RoBERTa employs dy-

dynamic masking, randomly selecting masked tokens per training example. These enhancements contribute to RoBERTa's improved capabilities without straying from the core transformer-based model design shared with BERT [9]. Specifically, in our implementation, we used 'all-distilroberta-v1' model of hugging face in OOV handling algorithm to generate embeddings. This model maps sentences paragraphs to a 768 dimensional dense vector space. This model is based on 'distilroberta-base' model, having 6 layers, 768 dimensions and 12 heads, and 82M parameters.

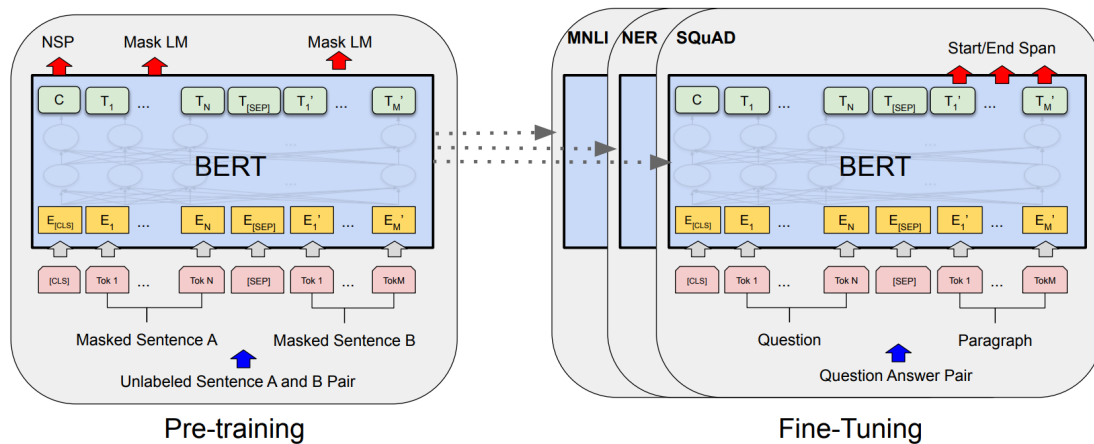


Figure 5.6: RoBERTa follows BERT architecture [4] with some additional changes

5.5 Results

Table 5.4 shows the comparison of scores between Personalization vs. Accuracy w.r.t user profiles models.

ROUGE score, which is an accuracy-based measure used in the paper. ROUGE-1, ROUGE-2, and ROUGE-L scores are as per mentioned below[1], where ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is a recall-based measure that gives a score between 0 and 1 based on the similarity between summaries, i.e., overlapping of n-grams in reference and model-generated summary[7]. The ROUGE-L metric calculates the ratio of the sum of the lengths of the LCSs between candidate and reference summaries to the sum of the lengths of the reference summaries. It emphasizes recall by considering the longest shared sequences of words between the candidate and reference summaries. By doing so, it aims to capture the overall content overlap and capture the important information conveyed in the reference summaries [7]. Formula of ROUGE-L is as per the following.

$$ROUGE - L = \frac{\sum_m \sum_n longest_common_subsequence(m,n)}{\sum_m \sum_n length(n)}$$

In the above formula, m represents a candidate summary, n represents a reference summary, *longest_common_subsequence(m,n)* calculates the length of the longest common subsequence between m and n, and *length(n)* represents the length of the reference summary n. The formula calculates the ratio of the sum of the lengths of the longest common subsequences between candidate and reference summaries to the sum of the lengths of the reference summaries.

When personalized measure EGISES is high, it indicates the model is highly insensitive, and a low ROUGE score indicates less overlapping between distribution of user and model summary, both of these indicate bad score, shown by the red color while green indicates good score.

| User embedding used | Injection type | EGISES | ROUGE 1 | ROUGE 2 | ROUGE L |
|---------------------|----------------|---------|---------|---------|---------|
| BRIO [8] | - | 0.65198 | 47.78 | 23.55 | 44.57 |
| PENS + NAML [15] | Type 1 | 0.89908 | 27.49 | 10.14 | 21.62 |
| PENS + NRMS [16] | Type 1 | 0.91642 | 26.15 | 9.37 | 21.03 |
| PENS + EBNR [12] | Type 1 | 0.95308 | 25.13 | 9.03 | 20.73 |
| PENS + EBNR [12] | Type 2 | 0.99513 | 25.49 | 9.14 | 20.82 |
| PENS + NRMS [16] | Type 2 | 0.99714 | 25.41 | 9.12 | 20.91 |

Table 5.4: Personalization vs. Accuracy w.r.t user profiles models

CHAPTER 6

Robustness of EGISES

EGISES' reliability demonstrated via ROUGE correlation, which can be considered as an indirect way to show correlation with human, since ROUGE score is a widely used metric with a high correlation with human judgment compared to other measures of accuracy.

6.1 Correlation between ROUGE-L Dev and DINS Dev

Since ROUGE-L score gives a score based on the similarity of generated summary with the reference summary, we used $1 - \text{ROUGE L}$ to get the distance between the user pair and the summary pair, which was previously calculated using the deviation function.

Specifically, we took the proportional difference between ROUGE, where the smaller value among user and summary distance will be the numerator, and the other one will be in the denominator. Since ROUGE is not commutative, we calculated the distance between u_1 and u_2 and vice versa and then averaged them out. Likewise, in deviation, where formulation itself includes a proportional difference between the user and summary pair where distance is calculated using weighted JSD. While calculating the deviation of the user pair, we took the assumption that only these 2 users exist in the system, that is, we just consider the deviation of that two users and their summary distance, not all rest of the users or summaries of that document. Then we calculated the correlation of user pair's ROUGE-L Dev and DINS Dev. The formula for the same can be seen in fig. 6.1.

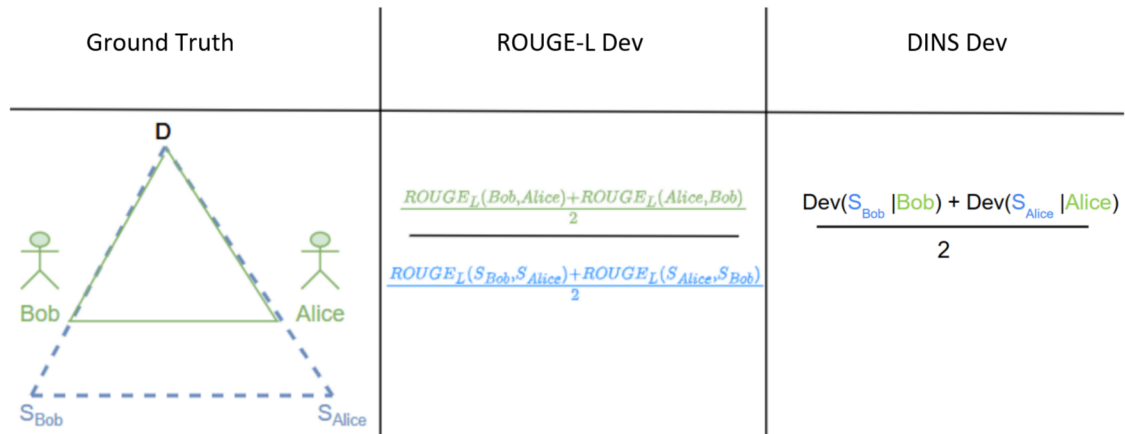


Figure 6.1: ROUGE-L Dev and DINS Dev

We calculated the correlation for the PENS+NAML user embedding[15] and injection type 1 model, which received the highest EGISES score. We determined the correlation coefficients using Pearson, Kendall, and Spearman.

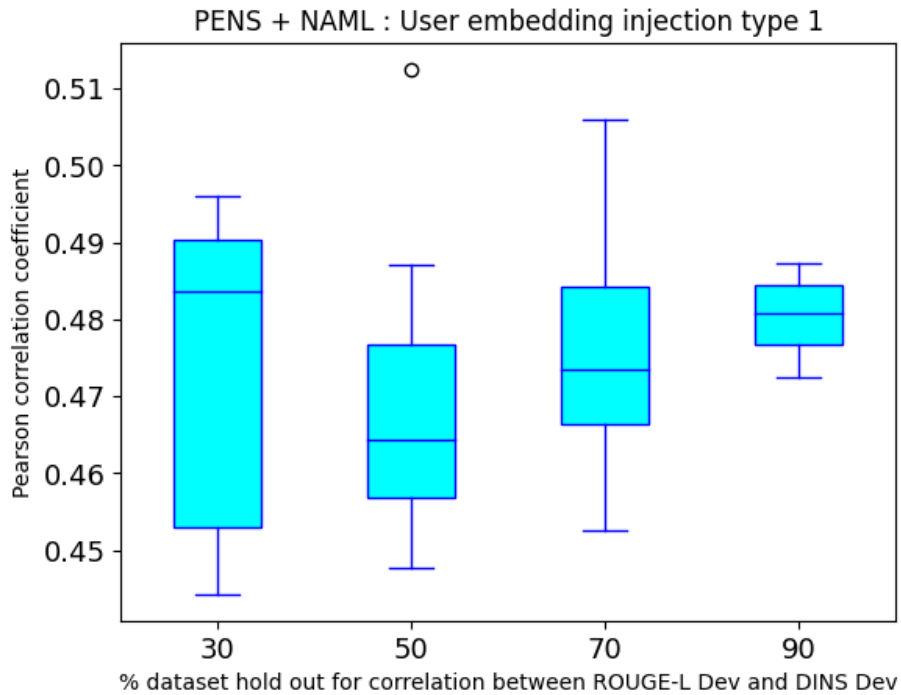


Figure 6.2: Correlation with ROUGE-L Dev and DINS Dev using Pearson

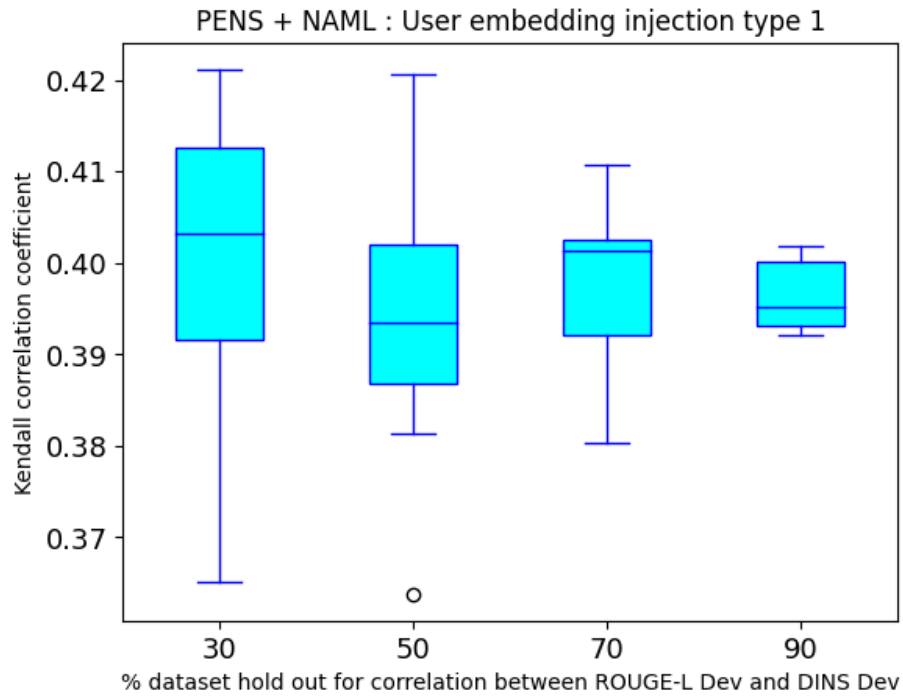


Figure 6.3: Correlation with ROUGE-L Dev and DINS Dev using Kendall

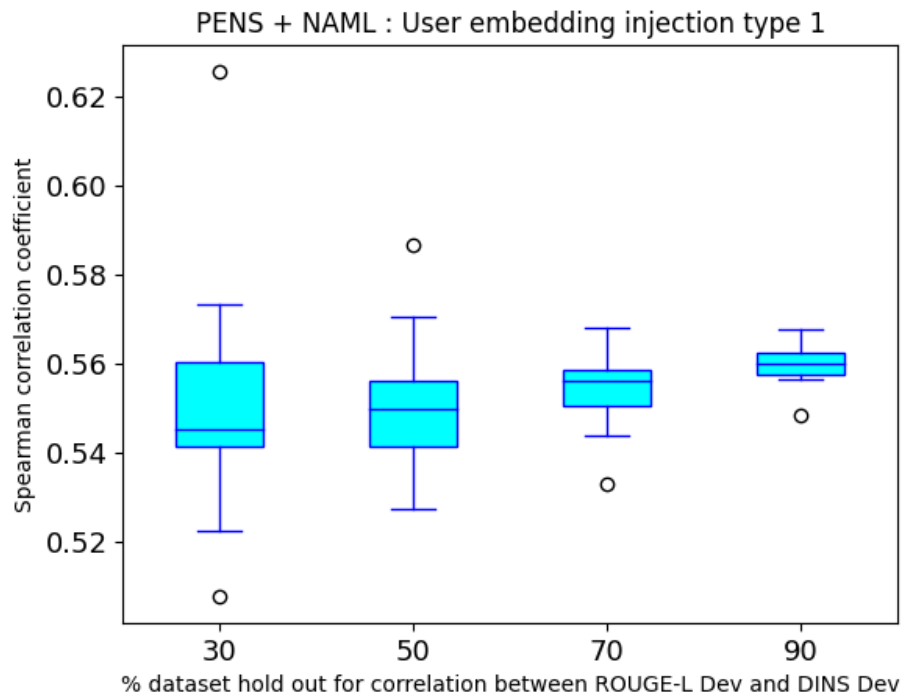


Figure 6.4: Correlation with ROUGE-L Dev and DINS Dev using Spearman

6.2 Correlation between ROUGE L and Personalized ROUGE

We propose Personalized ROUGE (P-ROUGE), a novel ROUGE based measure that not gives a score only basis on accuracy, rather also considers the degree of personalization while giving score.

Intuitively, we can consider $P - ROUGE = ROUGE_Score * ROUGE_Unit$. In this case, when unit is 1, it'll act as just ROUGE score. We formulated unit such a way it'll penalize ROUGE score if EGISES is high, which means model is highly insensitive. The mathematical formulation is as follows.

$$P - ROUGE = ROUGE_Score * ROUGE_Unit$$

where,

$$ROUGE_Unit = 1 - (\alpha * sigmoid((\beta * EGISES) / ROUGE_Score))$$

Here, $\alpha \in [0, 1]$ is the compensation coefficient and $\beta \in (0, 1]$ is personalization coefficient. In case of $\beta = 0$ excluded since it gives undesired result. For this formulation, we used ROUGE but any accuracy based measure can be used instead of just ROUGE.

Let's consider the PENS+NAML [15] model with user embedding injection type 1 as an example, whose initial ROUGE L score is 21.62. However, based on the model's EGISES, P-ROUGE is 12.17 as a penalty applied to the ROUGE score via the ROUGE_Unit. Likewise, PENS+EBNR [12] model with user embedding injection type 2, whose ROUGE L score is 20.82 and P-ROUGE is 12.15.

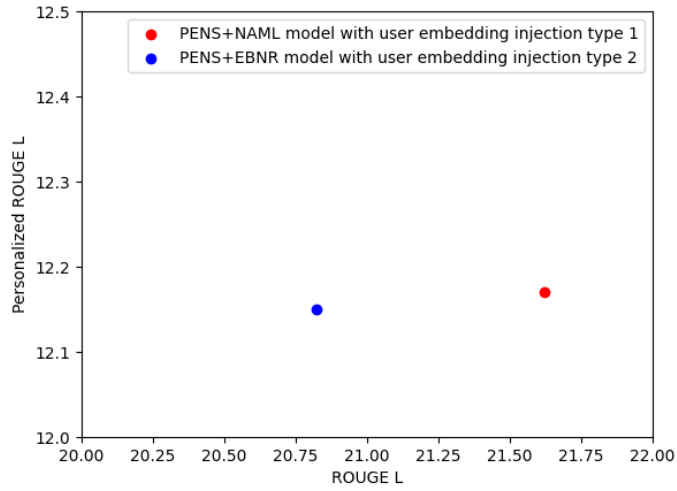


Figure 6.5: ROUGE L vs Personalized ROUGE L

Co-relation between ROUGE and Personalized ROUGE of PENS + NAML : User embedding injection type 1 model is as follows.

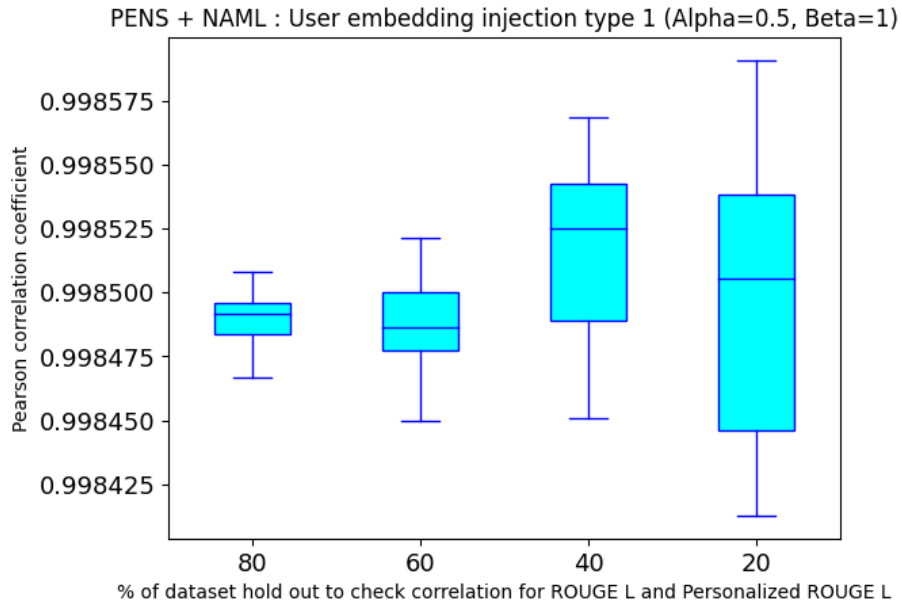


Figure 6.6: Correlation with ROUGE L and Personalized ROUGE L using Pearson

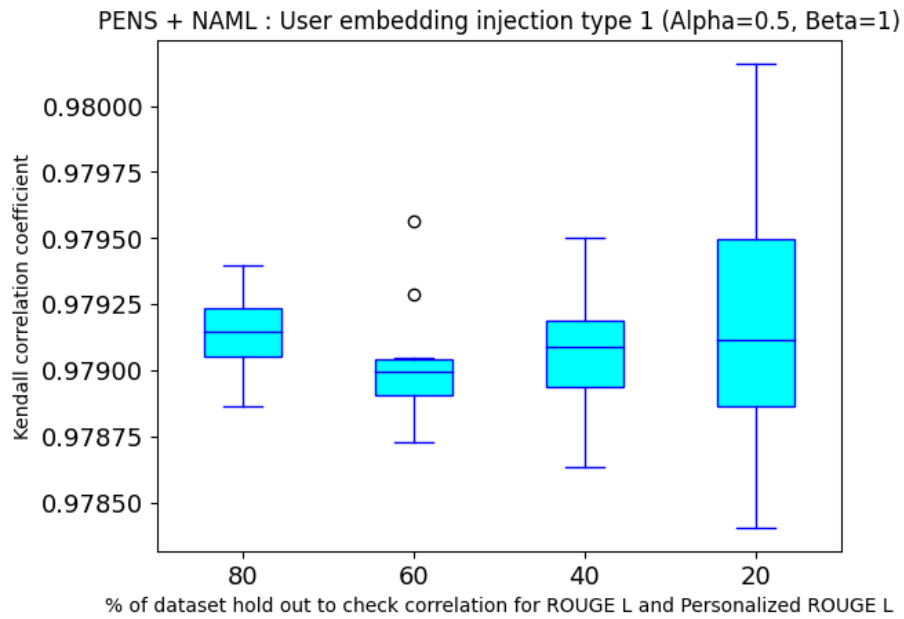


Figure 6.7: Correlation with ROUGE L and Personalized ROUGE L using Kendall

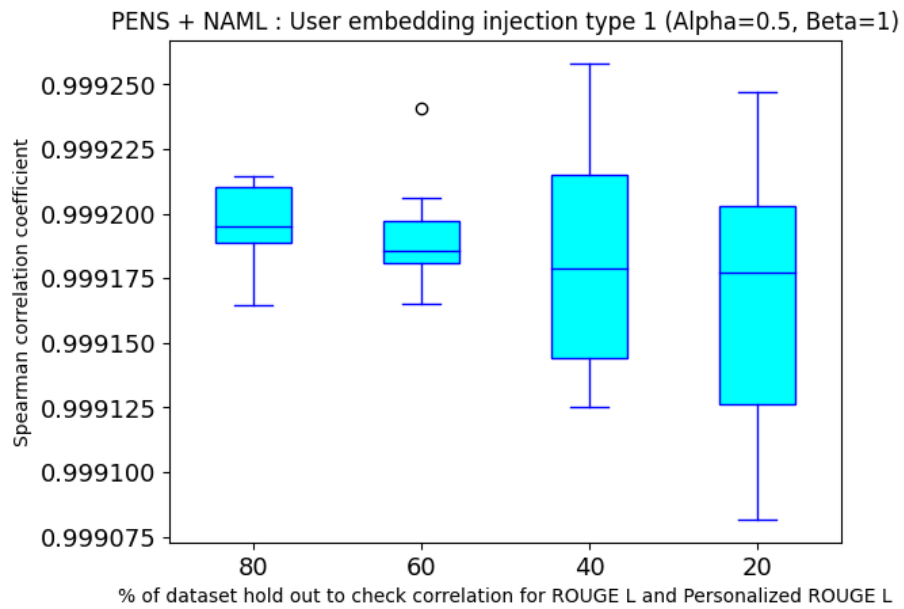


Figure 6.8: Correlation with ROUGE L and Personalized ROUGE L using Spearman

CHAPTER 7

Conclusions and future direction

Through our experiments, we can conclude that accuracy measures are not enough to measure the degree of personalization of the personalized summarization models. Hence, we propose EGISES to measure that. Additionally, we propose P-ROUGE that considers both, personalization and accuracy to generate the score. Further, using correlation with ROUGE we are able to show that our is reliable to use.

In future work, we are going to work on a direct way to get user agreement, that is gathering online survey responses to show direct reliability through human score correlation.

Since it's not practical to ask each user to know their agreement shown in 7.1, we can ask annotator give score based on similarity of summary pair7.2.

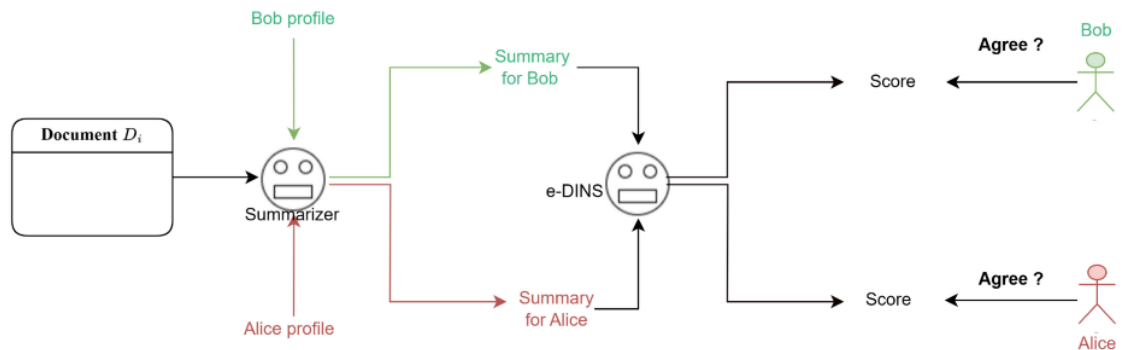


Figure 7.1: Direct agree meant of EGISES score by user

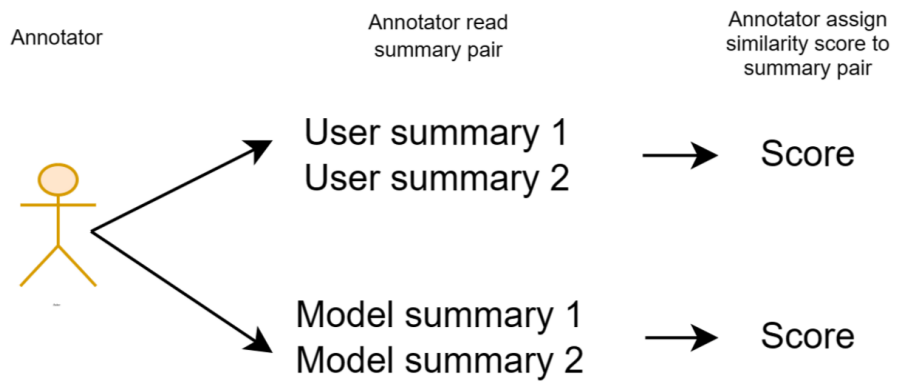


Figure 7.2: Annotator assign similarity score to summary pair

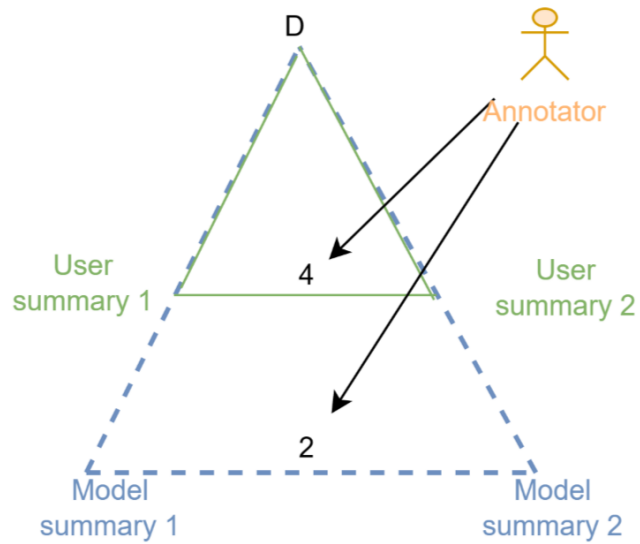



Figure 7.3: Illustration of score given by annotator

A summary pair can be model generated or user written, annotator will be not aware of it while assigning score on online survey as shown below 7.4.

 Dhirubhai Ambani
Institute of Information and Communication Technology

Evaluation Metric Correlation Survey

You are supposed to rate the summary pair based on similarity. The meaning of each score is given below.
1: Very similar, 2: Similar, 3: Somewhat similar, 4: Somewhat dissimilar, 5: Dissimilar, 6: Very dissimilar

Your Name (optional) _____

1st summary: powerball's june 29th drawing lands jackpot worth more than \$137m
2nd summary: the powerball jackpot is at 137 million

1 2 3 4 5 6

1st summary: powerball winning numbers for 6 29 2019 drawing
2nd summary: powerball winning numbers for 6 29 2019 drawing

1 2 3 4 5 6

1st summary: powerball winning numbers for 6 29 2019 drawing
2nd summary: powerball winning numbers for 6 29 2019 drawing

1 2 3 4 5 6

1st summary: powerball winning numbers for 6 29 2019 drawing
2nd summary: powerball winning numbers for 6 29 2019 drawing

1 2 3 4 5 6

1st summary: powerball numbers
2nd summary: powerball jackpot numbers

1 2 3 4 5 6

1st summary: powerball numbers
2nd summary: california, florida and tennessee reach largest jackpot in history

1 2 3 4 5 6

Disclaimer: This data is solely for research purpose. You may optionally add your name, which will be added to our contributor list when this dataset will be published.

Figure 7.4: Online survey setup

Other future works are as per following:

- Evaluating ChatGPT-styled models' degree of personalization w.r.t summarization [work-in-progress]
- Exploring SOTA High-Dimensional Contextual Vector based EGISES measures (instead of JSD) for measuring of personalization

To evaluate ChatGPT-styled models' degree of personalization w.r.t summarization, the setup will be as per following, depending on whether it is a prompt-based or instruction-based model.

- **Prompt-based models**

In the case of prompt based, where we give a set of documents and its personalized summary written by the user except the last document, we will ask the model to generate a personalized summary for the last document based on given examples. Likewise we can generate personalized summary for another user.

First, generate a summary of last document for user 1 by writing as per the following prompt. Where $\mathbf{d}_i^{U_j}$ is ith document shown to user j and $S_i^{U_j}$ is personalized summary written by user j for ith document

$$\begin{aligned} D_1^{U_1} &\longrightarrow S_1^{U_1} \\ D_2^{U_1} &\longrightarrow S_2^{U_1} \\ D_3^{U_1} &\longrightarrow S_3^{U_1} \\ D_4^{U_1} &\longrightarrow ? \end{aligned}$$

Generate a summary of the last document for user 2 by writing as per the following prompt:

$$\begin{aligned} D_1^{U_2} &\longrightarrow S_1^{U_2} \\ D_2^{U_2} &\longrightarrow S_2^{U_2} \\ D_3^{U_2} &\longrightarrow S_3^{U_2} \\ D_4^{U_2} &\longrightarrow ? \end{aligned}$$

Now by using this pair of model-generated and user-written personalized summaries, we can find deviation.

- **Instruction-based models**

In the case of instruction based (dialogue based), where we give a document and ask to generate a summary, respond to this summary by giving user 1 has written summary, saying user 1 was expecting this summary. We keep repeating the same with different documents several times and finally ask the model to generate a summary for the next document.

User 1: Write the summary for document D_1

Bot: Summary of D_1 is

User 1: But user 1 was expecting this... : personalized summary written by user 1 for D_1

User 1: Write the summary for document D_2

Bot: Summary of D_2 is

User 1: But user 1 was expecting this... : personalized summary written by user 1 for D_2

User 1: What will be the personalized summary of D_3 for user 1?

Likewise, generate summary for user 2 as per the following:

User 2: Write the summary for document D_1

Bot: Summary of D_1 is

User 2: But user 2 was expecting this... : personalized summary written by user 2 for D_1

User 2: Write the summary for document D_2

Bot: Summary of D_2 is

User 2: But user 2 was expecting this... : personalized summary written by user 2 for D_2

User 2: What will be the personalized summary of D_3 for user 2?

Now by using this pair of model-generated and user-written personalized summaries, we can find deviation.

References

- [1] X. Ao, X. Wang, L. Luo, Y. Qiao, Q. He, and X. Xie. PENS: A dataset and generic framework for personalized news headline generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 82–92, Online, Aug. 2021. Association for Computational Linguistics.
- [2] L. A. Cabrera-Diego and J.-M. Torres-Moreno. Summtriver: A new trivergent model to evaluate summaries automatically without human references. *Data Knowledge Engineering*, 113:184–197, 2018.
- [3] 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, 2023.
- [4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [5] S. Ghodratnama, A. Beheshti, M. Zakershaharak, and F. Sobhanmanesh. Extractive document summarization based on dynamic feature space mapping. *IEEE Access*, 8:139084–139095, 2020.
- [6] S. Ghodratnama, M. Zakershaharak, and F. Sobhanmanesh. Adaptive summaries: A personalized concept-based summarization approach by learning from users’ feedback. *CoRR*, abs/2012.13387, 2020.
- [7] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [8] Y. Liu, P. Liu, D. Radev, and G. Neubig. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland, May 2022. Association for Computational Linguistics.

- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.
- [10] M. Menéndez, J. Pardo, L. Pardo, and M. Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997.
- [11] K. M. Nenkova Ani, Passonneau Rebecca and S. Sigelman. Applying the pyramid method in duc2005.
- [12] S. Okura, Y. Tagami, S. Ono, and A. Tajima. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17*, page 1933–1942, New York, NY, USA, 2017. Association for Computing Machinery.
- [13] 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, 2022.
- [14] K. Veningston, P. V. Venkateswara Rao, and M. Ronalda. Personalized multi-document text summarization using deep learning techniques. *Procedia Computer Science*, 218:1220–1228, 2023. International Conference on Machine Learning and Data Engineering.
- [15] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie. Neural news recommendation with attentive multi-view learning, 2019.
- [16] C. Wu, F. Wu, S. Ge, T. Qi, Y. Huang, and X. Xie. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6389–6394, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.
- [17] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2020.
- [18] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online, July 2020. Association for Computational Linguistics.