Enhancing Argument Summarization: Prioritizing Exhaustiveness in Key Point Generation and Introducing an Automatic Coverage Evaluation Metric

Mohammad Khosravani[1], **Chenyang Huang**[2], **Amine Trabelsi**[3]
[1]Dept. of Computer Science, Lakehead University
[2]Amii, Dept. of Computing Science, University of Alberta
[3]Dept. of Computer Science, Université de Sherbrooke
mkhosrav@lakeheadu.ca chenyangh@ualberta.ca amine.trabelsi@usherbrooke.ca

Abstract

The proliferation of social media platforms has given rise to the amount of online debates and arguments. Consequently, the need for automatic summarization methods for such debates is imperative, however this area of summarization is rather understudied. The Key Point Analysis (KPA) task formulates argument summarization as representing the summary of a large collection of arguments in the form of concise sentences in bullet-style format, called key points. A sub-task of KPA, called Key Point Generation (KPG), focuses on generating these key points given the arguments. This paper introduces a novel extractive approach for key point generation, that outperforms previous state-of-the-art methods for the task. Our method utilizes an extractive clustering based approach that offers concise, high quality generated key points with higher coverage of reference summaries, and less redundant outputs. In addition, we show that the existing evaluation metrics for summarization such as ROUGE are incapable of differentiating between generated key points of different qualities. To this end, we propose a new evaluation metric for assessing the generated key points by their coverage. Our code can be accessed online.[1]

1 Introduction

Controversial issues that divide people are prevalent on social media platforms. These platforms provide fertile ground for individuals to express their arguments and engage in debates, leading to a vast amount of textual opinions expressed in diverse manners. Whether it's policymakers, businesses, or individuals, every decision-maker can benefit from a synthesized version that summarizes the main distinct key points related to an issue.

A key point (KP) should be a concise, high-level argument that aligns with a significant facet of a recurrent argument, while still being informative. Consider the topic of child vaccination. A valid key point might be, "Routine child vaccinations, or their side effects, are potentially dangerous.". Another distinct key point could be, "The decision of vaccinating a child should rest with parents, not the state." In this study, we aim to group a multitude of diversely expressed and sometimes poorly formulated raw arguments based on their facets, and to extract those that correspond to valid key points for each facet. Consequently, the summary, i.e., the collection of extracted arguments, should not contain redundancy and should cover the primary key points relevant to the topic of interest.

The work of Bar-Haim et al. (2020a) and Friedman et al. (2021) were the first to formulate the Key Point Analysis (KPA) task and to introduce a corresponding dataset, the ArgKP dataset. The KPA shared task was comprised of two tracks, Key Point Matching (KPM) and Key Point Generation (KPG). The goal of KPM is to map the correct key point to arguments, i.e. predicting whether argument-key point pairs match or not. In KPG, the goal is to generate the key points, given the arguments as inputs. In the KPA task, the term "generated key points" refers to output summary generated by the model. The model itself, whether abstractive or extractive, is referred as a KPG model (Friedman et al., 2021).

The top two previous approaches to KPG, as evaluated by the task organizers, were Bar-Haim et al. (2020b) and Alshomary et al. (2021). However, due to their construction that prioritizes the most popular recurrent arguments, both approaches often generate text that emphasizes *redundant* key points. Additionally, both methods rely on filtering low-quality arguments, which limit the input. This compromises the potential *coverage* of most or all key points, as less popular ones have a limited distribution, and further filtering may further diminish their chances of representation. Moreover, Bar-Haim et al. (2020b) select only short arguments for inclusion in a summary. This of-

[1] github.com/b14ck-sun/arg-sum

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8209–8221
June 16-21, 2024 ©2024 Association for Computational Linguistics

ten yields only a few arguments from an input of around two hundred and produces no summary for inputs of a hundred or fewer.

In this work, we introduce a novel and simple framework for key point generation. It is versatile, catering to both small and large numbers of input arguments. Most critically, it generates a summary comprised of self-contained arguments that cover the majority of existing key points (in a reference summary) with minimal redundancy. In this work we use key point generation and argument summarization interchangeably as the goal of the KPG model is to summarize a group of arguments.

On another aspect, the KPA task organizers relied on human judges to score and rank the KPG models. Judges were instructed to score output summaries based on their coverage of reference summaries and redundancy. Although human evaluation is accurate, it is neither scalable nor reproducible. Alternatively, automatic evaluation metrics, such as ROUGE (Lin, 2004), are often used for evaluating generated summaries. However, as shown by our experiments, they are not as effective for evaluating KPG task. Hence, in addition to the argument summarization approach, we propose a new evaluation metric for coverage based on the KPA shared task.

For the **summarization of arguments**, we employ an extractive clustering based approach. Using semantic-based clustering, we group similar arguments together and use a matching model to find the argument that best represents the cluster. The representatives of clusters are chosen as the summary. This approach allows less popular key points, i.e. smaller clusters, to be represented. Moreover, the matching model used in argument selection eliminates the need for a filtering step that not only limits the arguments that can be extracted, but also introduces additional error and significantly slows down the process. We also demonstrate the versatility of our model by comparing it to previous approaches on datasets other than the traditional ArgKP, which the earlier methods primarily utilized.

To **automatically assess the coverage** of argument summaries, we utilize a key point matching (KPM) model to compute a coverage score. This model determines whether a given produced argument matches with a reference or ground truth key point. The coverage score quantifies the proportion of matched reference key points in the summary. Furthermore, by enforcing a limit on the number of generated key points, a summary with high coverage of distinct key points would indicate low redundancy. Our experiments confirm the efficacy and appropriateness of the proposed metric compared to traditionally employed ones.

In short, our contributions are as follows:

1. A novel and simple extractive framework for argument summarization based on clustering, that generates concise high quality summary with more coverage and less redundancy compared to existing state-of-the-art work.

2. An evaluation metric for assessing summary coverage of the KPG task, that better correlates with the actual key point coverage of summaries compared to existing metrics.

2 Related Works

2.1 Argument Summarization Prior to the KPA Task

Prior to the introduction of the KPA task, the field of argument summarization was rather underdeveloped, both in terms of available datasets and techniques. However, several research focused on related experiments, such as clustering or extraction of arguments.

Misra et al. (2016) aimed to extract different aspects of arguments, called argument facets, similar to key points. The proposed framework first extracted the sentences that include arguments, and further ranked extracted sentences by their similarity to each other, with similar arguments representing an argument facet. Trabelsi and Zaïane (2019) developed an approach for unsupervised extraction of argument facets as phrases. These were used to derive arguments articulated in full sentences that reflected contrasting viewpoints or stances. They proposed a phrase topic model that leverages reply-interactions in online debates, enabling the effective clustering of viewpoints and the organized presentation of arguments. Ajjour et al. (2019) focused on frame identification using clustering, where a frame refers to arguments that cover the same aspect. The method first clusters the arguments into topics then removes topic-specific features from the arguments, and lastly clusters the "topic-free" arguments into frames. Reimers et al. (2019) similarly focuses on argument clustering and classification, however unlike the previous approach they use BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018) instead of TF-IDF and

LSA (Deerwester et al., 1990). The authors experimented with different classification and clustering methods in supervised and unsupervised settings.

2.2 Argument Summarization Following the Introduction of the KPA Task

The KPA task was based on Bar-Haim et al. (2020a)'s work which introduced a dataset of arguments and their reference summaries in the form of key points. The authors showed that a concise list of key points can cover the majority of the arguments covered. Bar-Haim et al. (2020b) and Alshomary et al. (2021) were the top performing models in the KPA task for both key point matching and key point generation tracks.

Our approach is inspired by Bar-Haim et al. (2020b) where the authors propose a ranking-based extractive method for key point generation. Specifically, first quality candidates are extracted from the input arguments. Next the candidates are ranked by the number of arguments they match to using a key point matching model. Lastly, candidate arguments are selected as summary if their similarity score with previously selected arguments is below a certain threshold. Similarly, we also select the highest coverage arguments as summaries, however, our method differs in the candidate selection, as we don't rely on selecting high-quality arguments as candidates.

The other top-performing model in the KPA task was proposed by Alshomary et al. (2021), which proposed an extractive method using PageRank, where the nodes are arguments and edges are the matching score between argument pairs. The mentioned approach removes low-quality arguments in the input and only establishes an edge if the matching score is above a threshold. Top nodes according to their importance score are selected as the reference summary, given their similarity score with previously selected nodes is below a threshold.

There has also been a number of research on key point generation following the introduction of the task. Li et al. (2023) propose a two-step abstractive summarization method based on clustering that utilizes large language models. However, unlike our approach that extracts the best argument in each cluster, they input all arguments belonging to a cluster into a language model for summarization. The proposed method uses BERTopic (Grootendorst, 2022) for clustering, specifically it uses a pre-trained language model (Reimers and Gurevych, 2019a) for getting contextualized embeddings, performs dimensionality reduction using UMAP (McInnes et al., 2018), and applies HDBSCAN (McInnes et al., 2017) to cluster embeddings. In the second step, it further fine-tunes an 11B parameter pre-trained language model (Chung et al., 2022) to generate the final summary, which is resource-intensive during both training and inference phases.

3 Methodology

3.1 Key Points Generation Models

This section describes the proposed framework and our approach to KPG. The input corpus consists of a list of arguments related to a specific topic, such as "Routine child vaccinations should be mandatory." Unlike the original task, the input data doesn't provide any stance information on an argument w.r.t. the topic (e.g., support or oppose). Having this stance information would aid in initially grouping arguments based on their stance, facilitating the separation of lexically similar key points with contrasting viewpoints. However, stance information is often absent in real-world social media texts, and our goal is to devise an effective method for this scenario. In order to generate summaries, our model first clusters the input arguments, and chooses the best representative sentence in each cluster as a candidate. The collection of candidates representing clusters is referred to as the generated key points (i.e. output summary). Figure 1 shows an overview of the approach.

3.1.1 Clustering

Previous approaches to KPG such as Bar-Haim et al. (2020b) and Alshomary et al. (2021) generate summaries from the input arguments, by favoring popular aspects. This leads to generated summaries that often over-represent some key points, while not mentioning others. This phenomenon occurs because there is often an imbalance in the number of arguments discussing different aspects or key points of a topic. For example, on the topic of "Routine child vaccinations should be mandatory", there is a total of nine reference or ground truth key points, in the ArgKP dataset (Bar-Haim et al., 2020a), where the most popular key point, has fifty arguments discussing it, whereas the least popular key point has only four. Our approach aims to represent all the key points equally, prioritizing but not favoring popular ones. This generates a summary with a higher coverage of reference key

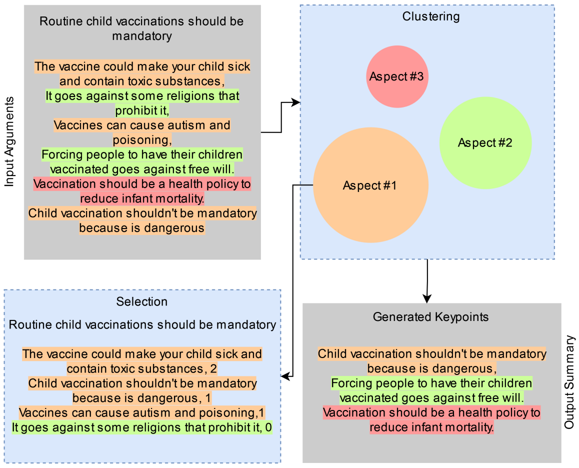
Input Arguments
Routine child vaccinations should be mandatory
The vaccine could make your child sick and contain toxic substances,
It goes against some religions that prohibit it,
Vaccines can cause autism and poisoning,
Forcing people to have their children vaccinated goes against free will.
Vaccination should be a health policy to reduce infant mortality.
Child vaccination shouldn't be mandatory because is dangerous
Clustering
Aspect #3
Aspect #2
Aspect #1
Selection
Routine child vaccinations should be mandatory
The vaccine could make your child sick and contain toxic substances, 2
Child vaccination shouldn't be mandatory because is dangerous, 1
Vaccines can cause autism and poisoning,1
It goes against some religions that prohibit it, 0
Generated Keypoints
Child vaccination shouldn't be mandatory because is dangerous,
Forcing people to have their children vaccinated goes against free will.
Vaccination should be a health policy to reduce infant mortality.
Output Summary

Figure 1: Visual depiction of the proposed KPG method. The color of arguments represents their key point or aspect. First the clustering step groups similar arguments. Next, the selection step chooses the argument with the highest score within each cluster as the cluster representative. The final summary aggregates the representatives from each cluster

points and a lower redundancy between generated arguments.

To achieve this goal, we cluster arguments, i.e. sentences, discussing similar key points together. To represent the arguments, we use Sentence-BERT (Reimers and Gurevych, 2019b) to generate embeddings. Then, we employ Agglomerative Clustering (AC) (Day and Edelsbrunner, 1984) to group them. We found that even the best performing pre-trained Sentence-BERT model (all-mpnet-base-v2) used for embedding leads to low Rand index scores, which means the clustered embeddings do not represent the actual clusters. To improve this, we fine-tuned the Sentence-BERT model on the ArgKP dataset using both contrastive loss following Alshomary et al. (2021) and Cosine Similarity loss. We used argument-key point pairs in ArgKP dataset as sentence pairs, and their matching label as labels for fine-tuning the Sentence-BERT model.

We observed that fine-tuning the language model improves the Rand index score in the clustering step by more than a 100%, and that the model fine-tuned using Cosine Similarity outperformed the one fine-tuned using Contrastive loss (Appendix B). We selected Agglomerative Clustering based on its superior Rand index scores. Additionally, Reimers et al. (2019) identified it as the optimal clustering algorithm for argument clustering and classification. Notably, Agglomerative Clustering doesn't require specifying the number of clusters as a hyperparameter, which is advantageous since this information is frequently unknown during inference. Nevertheless, Agglomerative Clustering remains adaptable to cases where the number of clusters can be estimated.

3.1.2 Selection

After clustering the arguments, we select one argument from each cluster that best represents the entire cluster (see Algorithm 1). This step is necessary to ensure that the selected arguments cover as many key points as possible. Ideally, the clustering step should cluster all arguments regarding a key point together. In which case, selecting an arbitrary argument from each cluster would be a true representation of that cluster. However, the clustering step does not reflect the actual or correct clusters perfectly. Hence, we need to select the most representative argument from each cluster. To achieve this, we sort the clusters by size to prioritize the more popular topics. This approach facilitates the extraction of the most prominent arguments first, from the top clusters, especially when the number of output arguments is constrained or limited to a specific count k.

To select the most representative argument from each cluster, a **Key Point Matching (KPM)** model, a fine-tuned BERT model in our case, is employed to predict matches between all possible argument-pairs within each cluster. The argument with the highest number of matches in each cluster is selected as the representative (See Algorithm 1 and Algorithm 2). In the case of a tie in the number of matches (i.e. when the matching model finds more than one argument that best represents the cluster) we select the shortest argument to prioritize conciseness. The mentioned BERT model is fine-tuned for entailment on the ArgKP dataset, where argument-key point pairs are given as the input, and the matching label is used as the class (0 for non-matching and 1 for matching). During inference, argument-key point pair are considered matching if the predicted label is 1 (Appendix C). A match between two arguments signifies that one argument entails the other. Consequently, the argument with the highest number of matches in each cluster, i.e. the representative, entails the most number of arguments inside its cluster. We refer to this approach as **Selection with Matching Model** (**SMM**).

While the generated summaries using the above selection method offer a high coverage of reference summaries, they often included long arguments, lengthening the overall summary. To this end, we, propose an alternative selection method, **Selection with Scoring Function** (**SSF**), where a specific scoring function is used. It takes both the coverage and overall length of an argument into account. The scoring function gives a score to an argument arg in each cluster c according to the number of matches while penalizing lengthy sentences as presented in line 5 of Algorithm 2. It uses i an exponent hyperparameter which determines the importance of matches. It controls the scoring function's emphasis on shorter sentences when candidates have similar match numbers.

Algorithm 1 Argument Selection

```
Input: Clusters $C$ (sorted largest to smallest),
Selection Function (SMM or SSF): $SF$,
Output: Generated Key Point $GKP$
for $c \ in \ C$ do
    for $arg \ in \ c$ do
        $c' \leftarrow c.remove(arg)$
        $Score_{c,arg} \leftarrow SF(c', arg)$
    end for
    $Selected_c \leftarrow argmax_{arg}(Score_{c,arg})$
end for
$GKP \leftarrow [Selected_{c1}, \dots, Selected_{|C|}]$
return $GKP$
```

Algorithm 2 Selection Function: SMM & SSF

```
    Input: Cluster $c$, Argument $arg$,
    Exponent Hyperparameter $i$,
    Key Point Matching Model: $KPM$
    Output: $Score_{c,arg}$
 1: procedure SMM($c, arg$)
 2:     return $Score_{c,arg} \leftarrow$ MATCHES$(c, arg)$
 3: end procedure

 4: procedure SSF($c, arg$)
 5:     return $Score_{c,arg} \leftarrow \frac{\text{MATCHES}_{(c,arg)}^{i}}{\#ofwords(arg)}$
 6: end procedure

 7: procedure MATCHES($c, arg$)
 8:     $matches \leftarrow 0$
 9:     for $arg' \ in \ c$ do
10:         if $KPM(arg, arg')$ then
11:             $matches++$
12:         end if
13:     end for
14:     return $matches$
15: end procedure
```

3.2 Coverage Evaluation Measure

The standard evaluation measure for summarization tasks, ROUGE (Lin, 2004), is not effective for evaluating the KPG task. In addition to its reported problems on reproducibility and comparability (Grusky, 2023), ROUGE score is concerned with the n-grams overlap between candidates and summaries, and fails to consider the semantics of documents (Li et al., 2023). This makes the ROUGE score less effective for evaluating arguments where the same viewpoint can be expressed with completely different wording and opposing arguments can have similar words. In the KPG task, an ideal summary should be concise, non-redundant, and cover different aspects or facets of the topic, according to the task organizers. To this end, task organizers used human judges to evaluate the generated summaries based on how redundant a summary is, and how it captures central points to the topic. Inspired by this, we propose an automatic evaluation metric for the KPG task, a coverage measure. However, unlike the KPA task where the goal was to cover the most important key points, we argue that covering more aspects of a topic is more important as it leads to a more comprehensive summary. It is also equally important to have a summary with a low number of duplicates, i.e. arguments that cover the same aspect or key point. The coverage measure assesses the extent to which different aspects of the topic are represented.

When the number of generated key points (i.e. output arguments) in a summary is capped or limited to a ground truth number of arguments, a high key point coverage value signifies that most arguments are distinct, thereby implying minimal redundancy.

3.2.1 Coverage Computation

The coverage measure predicts the percentage of actual ground truth key points captured by the produced summary. Our proposed coverage metric computes a coverage score by pairing each generated key point with every reference key point and use a matching model to predict if the argument-key point pair is a match. The coverage score is calculated as the percentage of all reference key points covered. As for the matching model, any classification method, such as the ones submitted for the key point matching track of the KPA task can be used, since the goal is to predict matching argument-key point pairs. Additionally, the BLEURT and BARTScore models used by Li et al. (2023) also evaluate coverage since the proposed soft-precision metric matches each reference key point with the highest scoring generated key point (and vice-versa for soft-recall). Our proposed key point matching uses the same fine-tuned BERT model in Section 3.1.2, similar to Bar-Haim et al. (2020b). We frame the KPM problem as an entailment problem unlike some previous approaches that use cosine similarity between argument and key point embeddings. Details on implementation settings can be found in Appendix C.

3.2.2 Coverage Datasets

To test the performance of different evaluation metrics, we created a set of pseudo summaries from the ArgKP test set, with different levels of coverage, 100%, 75%, and 50%. We named them Coverage Datasets – a dataset for each level of coverage. These pseudo summaries each contain the same number of arguments i.e. 25, sampled from the unseen ArgKP test set. Each coverage dataset, only covers its respective percentage of key points from the ArgKP test set. For example, for the 50% coverage dataset, we randomly select 50% of the key points on each topic (e.g., child vaccination), and only include arguments from those key points, with each selected key point having at least one argument related to them. To ensure fairness, the total number of arguments in each dataset is equal. Additionally, each coverage dataset was randomly sampled ten times. As an example the 100% coverage dataset, is randomly sampled 10 times, each time with different 25 arguments that cover 100% of the key points. Any results reported for coverage datasets are averaged over all ten instances. We utilize these coverage datasets to evaluate the efficacy of various evaluation metrics by tasking them to estimate the coverage provided by these datasets.

4 Experimental Setup

4.1 Experiments on the KPG Method

Overview: We perform both automatic and human evaluations to prove the effectiveness of our method. To showcase the ability of the proposed method to generate superior summaries, we compare the generated key points to previous approaches. Mainly we compare our work to the top two performing models in KPA tasks, Alshomary et al. (2021) and Bar-Haim et al. (2020b)[2]. We experiment on both versions of our selection method discussed in Section 3.1.2: SMM and SSF. We evaluate and compare different methods based on their coverage, redundancy, conciseness, and argument quality. For comparison, we limit the number of generated outputs by the models to the number of key points in the reference (ground truth) datasets. This was done as different methods produce summaries with different numbers of key points, and longer summaries are more likely to have a higher coverage of reference key points. Additionally, we have grouped all arguments that do not belong to any existing key point in the ArgKP dataset under one shared key point. This ensures that arguments that do not belong to any key point also contribute to coverage score.

Coverage and Redundancy: In order to evaluate the actual coverage, we calculate the percentage of reference key points covered by the generated key points. As all the methods are extractive, the true key point for each generated key point is known, which allows us to calculate the effective and actual percentage of reference key points covered. It is important to note that in this experiment we calculate the actual coverage using labeled data, and not the predicted coverage computed by our proposed metric. For redundancy, we consider the generated key points that belong to the same reference key point as duplicates and count the number of duplicates in each model, and present the percentage of duplicates (number of duplicate pairs divided by the number of all possible pairs).

Conciseness and Quality: In order to judge the quality of generated key points, we asked human judges to score those produced by different methods with respect to their reference key points. While this evaluation might seem unnecessary since the proposed and previous approaches are extractive, these approaches modify the structure of arguments by splitting multi-sentence arguments into single sentences which could affect their clarity. Inspired by human evaluation done by Li et al. (2023), we ask the judges to score argument-key point pairs (+1 for matches, -1 nonmatches, 0 for not sure) based on two conditions: 1. The argument covers the same aspect as the key point. 2. The argument is a clear and understandable argument regarding an aspect by itself given the topic (complete instructions are provided in

[2] early-access-program.debater.res.ibm.com

ArgKP	Coverage	Redundancy	R1	R2	RL
Method (SMM)	**59.59%**	2.46%	0.152	0.026	0.132
Method (SSF)	57.67%	**2.27%**	0.158	0.026	0.141
Alshomary w SS	49.09%	3.41%	0.195	**0.032**	0.175
Alshomary	45.45%	3.23%	**0.202**	0.028	**0.186**
BarHaim	37.67%	5.26%	0.153	0.028	0.136

Debate	Coverage	Redundancy	R1	R2	RL
Method (SMM)	**72.38%**	**1.42%**	0.06	0.005	0.052
Method (SSF)	57.98%	1.86%	0.079	0.007	0.069
Alshomary	64.88%	1.68%	0.068	0.003	0.062
BarHaim	56.41%	1.74%	**0.084**	**0.015**	**0.08**

Table 1: Actual coverage, redundancy, and ROUGE scores for each model's output. The coverage and redundancy are computed using the labeled data. SS refers to separate stances. Numbers in bold represent the best results.

Appendix D). We paired each generated key point with its respective matching key point based on the ArgKP dataset. For example, on the topic "Is US a good country to live in?" the generated key point "it doesn't provide it's citizens with free healthcare", was paired with the reference key point "The US has unfair health and education policies".

We sampled a total of 20 generated key points from four topics, two topics from the Debate dataset (Section 4.3) and two from ArgKP, for each model. We asked three graduate students to score each pair and averaged the score across judges for each pair. Lastly, we compare the average number of words per generated key point over all topics in the ArgKP test set to showcase the ability of the proposed method to generate concise outputs.

4.2 Experiments on Evaluation Metrics

Coverage Prediction: We assess the performance of our coverage measure compared to other metrics for KPG, namely ROUGE and Li et al. (2023). We assessed the metrics by tasking the evaluation metrics and models to predict the coverage of different Coverage datasets (Section 3.2.2). This allowed us to assess how good the evaluation metrics are at correlating an effective high coverage and low redundancy summary to a high score. First, we computed the ROUGE score on the datasets, with the key points as reference summaries.

Second, we use the evaluation metric introduced by Li et al. (2023) for computing Soft-Precision and Soft-Recall. The proposed Soft-Precision aims to finds the reference key point with the highest similarity score for each generated key point and vice-versa for Soft-Recall. Soft-F1 is the harmonic mean between soft-precision and soft-recall. The similarity score is calculated by BLEURT and BARTScore models, given the generated and reference key points as inputs. The Soft-F1 aims to evaluate the validity, sentiment, informativeness, single-aspect, significance, **coverage**, faithfulness, and redundancy of generated key points. In our experiment, we assigned every generated key point to the reference key point with the highest matching score, similar to the authors' approach, which evaluates the effectiveness of models at correctly assigning key points to arguments. We compared the predicted coverage of each metric to the actual coverage.

On Generated Summaries: We also evaluated summaries generated by different models using ROUGE, soft-F1 scores Li et al. (2023), and the proposed coverage measure. However, due to Soft-F1 metric limitation, where the number of generated key points in the output should match the number of reference key points, we were not able to evaluate the outputs generated by every model.

4.3 Dataset

The experiments are done on the dataset proposed for the KPA task, the ArgKP dataset (Friedman et al., 2021) with 24K argument/key-point pairs. The dataset contains 24 topics for train, and 4 for development, with 800 arguments and 3 topics for test set. To further prove the effectiveness and generalizability of our approaches, we compared our method and evaluation metric on the Debate dataset (Hasan and Ng, 2014). The Debate dataset contains arguments and aspects, similar to the ArgKP dataset, on four topics, with 3K argument/key-point pairs. To assess our proposed argument summarization approach, we compared

ArgKP	Coverage Measure	BLEURT	BARTScore	R1	R2	RL
100%	**83.18%** $_{\pm 0.052}$	66.26% $_{\pm 0.006}$	60.29% $_{\pm 0.044}$	0.166 $_{\pm 0.007}$	0.032$_{\pm 0.005}$	0.150 $_{\pm 0.007}$
75%	**77.27%** $_{\pm 0.056}$	61.11% $_{\pm 0.015}$	59.26% $_{\pm 0.023}$	0.169 $_{\pm 0.010}$	0.033 $_{\pm 0.006}$	0.153 $_{\pm 0.010}$
50%	64.84% $_{\pm 0.055}$	**56.38%** $_{\pm 0.017}$	57.61% $_{\pm 0.029}$	0.165 $_{\pm 0.011}$	0.033 $_{\pm 0.006}$	0.151 $_{\pm 0.011}$

Debate	Coverage Measure	BLEURT	BARTScore	R1	R2	RL
100%	82.08% $_{\pm 0.067}$	91.67% $_{\pm 0.015}$	**92.71%** $_{\pm 0.015}$	0.064 $_{\pm 0.002}$	0.005 $_{\pm 0.000}$	0.056 $_{\pm 0.001}$
75%	**76.67%** $_{\pm 0.040}$	90.63% $_{\pm 0.015}$	91.67% $_{\pm 0.015}$	0.066 $_{\pm 0.003}$	0.006 $_{\pm 0.001}$	0.057 $_{\pm 0.002}$
50%	**70.42%** $_{\pm 0.063}$	89.58% $_{\pm 0.059}$	94.79% $_{\pm 0.044}$	0.064 $_{\pm 0.003}$	0.006 $_{\pm 0.001}$	0.056 $_{\pm 0.003}$

Table 2: Different metrics for coverage prediction on coverage datasets (100%, 75%, 50%). The numbers in bold represent the best results (closest prediction to actual coverage). Values in subscripts represent standard deviation.

it with Bar-Haim et al. (2020b), and Alshomary et al. (2021) on the Debate dataset. To assess our coverage evaluation measure, we compared it to BLEURT and BARTScore for coverage prediction. To this end, we generated a set of different coverage datasets from the Debate dataset, similar to the previous coverage dataset (Section 3.2.2) with 75 arguments for each dataset, as the Debate dataset has more arguments per topic.

5 Results and Analysis

5.1 Evaluation of the KPG Method

5.1.1 Coverage and Redundancy

Our method outperforms both Bar-Haim et al. (2020b) and Alshomary et al. (2021) methods in producing more exhaustive and less redundant key points on the ArgKP dataset. In addition, we compare the output of our model, with Alshomary's output on the original KPA task when arguments are separated by their stances (Alshomary w SS). However, we cannot apply BarHaim's method similarly, as it requires more than 100 arguments per input, and each set of arguments separated by stance has fewer. Table 1 shows the coverage, redundancy and the ROUGE score of the proposed method, BarHaim, and Alshomary on the ArgKP and the Debate datasets with respect to the reference (true) key points. Our experiments show that both proposed versions of our approach, SMM and SSF, outperform other methods on the ArgKP dataset. The proposed methods are also generalizable to the Debate dataset as the method with SMM also outperforms both other methods and the method with SSF outperforms BarHaim on the Debate dataset. It is also important to note that the inferior performance of BarHaim in ArgKP dataset is partly due to the fact that it has a maximum length limit on the extracted key points, which makes the model generate less key points. However, this limit is fine-tuned for this dataset and cannot be changed or modified. On another aspect, ROUGE scores show no correlation with coverage and redundancy of outputs. We examine ROUGE's performance in more detail in Sections 5.2.1 and 5.2.2.

5.1.2 Conciseness and Quality

To showcase the proposed method's ability to generate quality short sentences, we compare the average word length and the human-evaluated quality scores of generated key points. Table 4 shows the scores averaged over all outputs. The results show our method surpasses Alshomary's in conciseness, yet BarHaim's approach, which only extracts short arguments, still yields briefer key points.

Moreover, the human judges found the proposed methods' outputs more understandable, as the best with method with SMM having the highest score and method with SSF having a similar score to the Alshomary's method. The human scores also indicate that longer outputs are easier to understand. The Krippendorff's α for inter-annotator agreement is 0.47 across all topics, with a score of 0.53 on the ArgKP dataset, and 0.40 on the Debate dataset.

5.2 Evaluation of the Coverage Measure

5.2.1 Comparison of the Evaluation Metrics

We compared the performance of our coverage measure to ROUGE and the method proposed by Li et al. (2023). For comparison, we used the proposed coverage measure, BLEURT, and BARTScore, to predict the coverage of different coverage datasets. Additionally, we calculated the ROUGE score of coverage datasets to examine ROUGE's ability to score outputs of different quality but the same length. Table 2 shows the predicted coverage of datasets using all evaluation metrics.

ArgKP	Coverage	Redundancy	F1 BL	F1 BS	R1	R2	RL
Method (SMM)	35.75% (**59.59%**)	(2.46%)	49.01%	57.94%	0.152	0.026	0.132
Method (SSF)	**39.89%** (57.67%)	(**2.27%**)	52.84%	60.81%	0.158	0.026	0.141
Alshomary	30.70% (45.45%)	(3.23%)	**54.58%**	**63.04%**	**0.202**	**0.028**	**0.186**
BarHaim	23.53% (37.67%)	(5.26%)	N/A	N/A	0.153	**0.028**	0.136

Table 3: The predicted coverage of models, alongside the Soft-F1 scores using BLEURT (BL) and BARTScore (BS), and ROUGE. The numbers in parenthesis represent the actual coverage and redundancy. Numbers in bold represent the best output.

Method	Avg. Words	Arg. Quality
Method (SMM)	18.6	**0.63**
Method (SSF)	12.4	0.4
Alshomary	15.3	0.46
BarHaim	**6.8**	0.23

Table 4: Average number of words and quality scores per generated key point, averaged for each model.

Our experiments show that the ROUGE score is not capable of differentiating between datasets with different levels of coverage. The BARTScore performs only slightly better, as the difference in predicted coverages is insignificant. BLEURT, however, performs reasonably well on the ArgKP dataset, but it does not generalize as well to the Debate dataset. We hypothesize the ineffectiveness of BLEURT and BARTScore is due to the fact that they are not trained on argumentative text, which uses a different distribution of words. ROUGE's poor performance, on the other hand, is the result method's inability to take the semantics of sentences into account, this is especially true in the argument domain as arguments on the same topic belonging to different key points often use similar words, which is not handled effectively by ROUGE. The inaccurate coverage prediction of BLEURT and BARTScore indicates that the models are not capable of assigning the generated key points to the correct reference key points, making them ineffective at predicting soft-F1 score. In contrast, the proposed approach's predictions are closer to the actual coverage of datasets. Also, the metric's effectiveness extends to unseen datasets, as evidenced by its performance on the Debate dataset.

Lastly, our KPM model used in the coverage metric (see Section 3.2.1) demonstrates high accuracy on the ArgKP dataset. On the ArgKP test set, it scores a 90.01% micro percision, compared to the 87.79% of the best performing KPM model in the KPA task by Alshomary et al. (2021).

5.2.2 Evaluation of the Coverage Metrics on the Generated Outputs

We further assessed the generated outputs using ROUGE, our designated metric, and the metrics introduced by Li et al. (2023). Table 3 displays the scores predicted by each evaluation metric, alongside the actual coverage and redundancy values. The Soft-F1 score can only be calculated when the number of generated key points is equal to the number of reference key points, making it incompatible with BarHaim. Table 3's results suggest that our proposed coverage measure often estimates the relative ranking of outputs accurately. On the other hand, alternative evaluation metrics don't consistently link a high coverage/low redundancy output with a top score. Notably, both F1 measures identify Alshomary's output as the best among the evaluated outputs. However, according to Table 1 Alshomary's output has a lower coverage of reference key points and higher redundancy in generated key points. Also according to Table 4, it has lower argument quality and comparable length to method with SMM. In this case it is not clear how/why a certain summary is better than the others. Therefore, we argue that employing a single metric to compare generated key points overall is not effective or descriptive. This is because generated KPs vary in aspects like coverage, redundancy, length, and quality, each demanding its own evaluation.

6 Conclusion

This paper proposes a new extractive method for KPG that aims to generate key points with a high coverage of reference key points and less redundancy. Our method adopts a clustering-based approach with two distinct selection methods to achieve this goal. Our experiments show that the proposed methods outperform previous SOTA in terms of coverage, redundancy, and quality . Lastly, we proposed a metric for coverage, that outperforms previous metrics on coverage prediction.

Limitations

Dataset: The test set for ArgKP dataset, while relatively small, is the only dataset for the task. However we aimed to verify our experiments by utilizing another dataset. We also did not filter offensive arguments in the corpus, as a result the generated result may include text which some readers might find offensive. It is important to highlight that detecting inappropriate language wasn't the goal of this study.

Method & Evaluation Metrics: The evaluation metrics, while more accurate than previous approaches, are still not accurate enough to rank the models with 100% accuracy, when the actual coverage and redundancy of two generated summaries is insignificant. Additionally, the coverage evaluation metric does not factor in the hierarchical entailment relation between arguments themselves or arguments and reference key points. As an example, the model might predict that a general reference key point (e.g. vaccinations may have unwanted side effects) is covered by a specific generated key point (e.g. vaccinations may have a specific side effect). Prioritizing these general key points over specific ones could be a potential future work, which might also require a dataset where the hierarchical relations between arguments are traced or annotated (i.e. which key points cover more fine-grained ones).
Additionally, the proposed approach has no way of enforcing generation of relatively equal key points for both stances. This could potentially lead to summaries that favor one stance more often than the other.

Ethics Statement

The dataset used for training/fine-tuning models is an existing anonymized dataset without data protection issues. Moreover, the human annotators for the task were informed that their work would be used, and were unpaid volunteers unrelated to the research. Only trained annotators who understood the task were selected. The only collected data was ratings produced by the annotators.

Acknowledgements

This research was partially supported by the Vector Institute through the Vector Scholarship in Artificial Intelligence, by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN-2022-04789, and by Compute Ontario (https://www.computeontario.ca/) and the Digital Research Alliance of Canada (https://alliancecan.ca/). We would like to thank the annotators and the reviewers for their valuable feedback.

References

Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019. Modeling frames in argumentation. 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 2922–2932, Hong Kong, China. Association for Computational Linguistics.

Milad Alshomary, Timon Gurcke, Shahbaz Syed, Philipp Heinisch, Maximilian Spliethöver, Philipp Cimiano, Martin Potthast, and Henning Wachsmuth. 2021. Key point analysis via contrastive learning and extractive argument summarization. In *Proceedings of the 8th Workshop on Argument Mining*, pages 184–189, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020a. From arguments to key points: Towards automatic argument summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039, Online. Association for Computational Linguistics.

Roy Bar-Haim, Yoav Kantor, Lilach Eden, Roni Friedman, Dan Lahav, and Noam Slonim. 2020b. Quantitative argument summarization and beyond: Cross-domain key point analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 39–49, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

W.H.E. Day and H. Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. journal of classification.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the*

American Society for Information Science, 41(6):391–407.

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

Roni Friedman, Lena Dankin, Yufang Hou, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. Overview of the 2021 key point analysis shared task. In *Proceedings of the 8th Workshop on Argument Mining*, pages 154–164, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Max Grusky. 2023. Rogue scores. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1914–1934, Toronto, Canada. Association for Computational Linguistics.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762, Doha, Qatar. Association for Computational Linguistics.

Hao Li, Viktor Schlegel, Riza Batista-Navarro, and Goran Nenadic. 2023. Do you hear the people sing? key point analysis via iterative clustering and abstractive summarisation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14064–14080, Toronto, Canada. Association for Computational Linguistics.

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

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

Amita Misra, Brian Ecker, and Marilyn Walker. 2016. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287, Los Angeles. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Amine Trabelsi and Osmar R. Zaïane. 2019. Phaitv: A phrase author interaction topic viewpoint model for the summarization of reasons expressed by polarized stances. *Proceedings of the International AAAI Conference on Web and Social Media*, 13(01):482–492.

A Output Examples on Child Vaccination Topic

Outputs generated by BarHaim

- vaccinating children helps eradicate disease
- Child vaccination shouldn't be mandatory because is dangerous
- to keep schools safe children must be vaccinated
- protecting infants must be a priority for all
- this vaccine could cause unwanted side effects
- vaccination at birth is a human duty

Outputs generated by Alshomary

- People all around the world vaccinate their children to protect them from any life threatening disease.

- Parents should be allowed to choose if their child is vaccinated or not.

- Parents don't always know best and failure to vaccinate can be catastrophic for a child.

- child vaccinations should be mandatory to provide decent health care to all.

- Vaccines in children should not be mandatory because they can have consequences for their health in the future

- When vaccines are mandatory, they can infringe on family religious choices

- each child must be vaccinated so that they can live more peacefully

- childhood vaccination is necessary as it helps the growth of children

- child vaccinations is not mandatory because it may cost the country unnecessary funds.

Output generated by method with SMM

- vaccines must be compulsory for children because in this way we prevent later diseases in infants,

- Child vaccination should be mandatory because they may transfer the virus,

- someone from the child population could suffer side effects,

- No-one can tell a parent that they must vaccinate their child, it is against their human rights,

- Child vaccination shouldn't be mandatory because children don't catch the virus,

- Children are the future of the country and the world, taking care of them should be our greatest concern so that they grow up as healthy adults that is why vaccination should be mandatory,

- to keep schools safe children must be vaccinated,

- Vaccines save up to 3 million lives a year, protecting children from life-threatening and highly infectious diseases,

- Routine childhood vaccinations should not be mandatory, because the contraindications or what negative effect it may cause the child are not known, the parents should decide,

Output generated by method with SFM

- Prevents a large number of diseases,

- this vaccine could cause unwanted side effects,

- Parents should decide what is best for their child.,

- Child vaccination should be mandatory to avoid the virus,

- Child vaccination shouldn't be mandatory because children don't catch the virus,

- protecting infants must be a priority for all,

- to keep schools safe children must be vaccinated,

- because they can have very dangerous reactions to vaccines,

- the vaccine provide inmunity to the people and prevents to contract the dissease

B Rand Index Scores

Our experiments show that fine-tuning the SBERT model significantly improves the accuracy of clustering step, which improves the overall quality of generated key points. We use Rand index score to measure the accuracy of predicted clusters w.r.t to the original clusters. We also remove all arguments that do not belong to any clusters for this evaluation. Table below shows the Rand index score improvements after fine-tuning the model using both contrastive and cosine similarity loss.

Rand Index Score	Topic1	Topic2	Topic3
Before FT	0.18	0.19	0.15
FT w CS	0.43	0.45	0.39
FT w Cont.	0.38	0.44	0.32

Table 5: Rand index scores for the clustering before and after fine-tuning on the ArgKP train set using Cosine Similarity loss and Contrastive loss. Topic 1, 2, and 3 refer to topics in ArgKP test set.

C Implementation Details

C.1 Key Point Generation Models

For embedding generation, we use "all-mpnet-base-v2", and fine-tune it using Cosine similarity loss the ArgKP dataset. Agglomerative clustering with a distance threshold of 1.5 is used for clustering embeddings. For the argument selector, we used a fine-tuned BERT on the ArgKP dataset. We format each input as "[CLS] argument [SEP] key point [SEP]" for input, and [logit 1, logit 2], where [1, 0] represents class zero or non-matching and [0, 1] represents matching argument-key point pairs, for the output. A pair is considered matching if the second logit is greater than the first one. For the Selection with Scoring (SSF), we tried different formulas (e.g. the number of matches, MATCHES, divided by number of words or e^{MATCHES} divided by number of words). Empirically, we find that setting the exponent hyperparameter i to 5 (see Algorithm 2) produces the best result.

C.2 Evaluation Metric

We used the same BERT model used in argument selection for coverage, however we make a few modifications. First, we limited the number of reference key points that can be assigned to a generated key point to one. In case of multiple reference key points matching to a single generated key point, we assign the highest scoring reference key point to the generated one. Our experiments show that setting this limit of maximum one improves the accuracy. This is in line with the nature of input arguments as only 4% of the arguments cover more than one key point. We also experimented with setting different values for limits and thresholds on the coverage model similar to Bar-Haim et al. (2020b). As for the ROUGE metric, we used Python's rouge_score library version 0.1.2.

D Task Description for Human Evaluators

Task You are given a pair of argument and key point sentences, and your have to decide whether the pair match or not A matching pair should satisfy two conditions: 1. The argument should cover the same aspect as the key point. 2. The argument should be a clear and understandable argument regarding an aspect by itself given the topic. You are supposed to give a score to each argument key point pair Score 1 if the argument and key point are matching Score -1 if the argument and key point are not matching Score 0 if you are not sure **Data**

- You are given 80 sentences in total, 20 sentences for 4 topics. The topics are:
  - The USA is a good country to live in
  - Social media platforms should be regulated by the government
  - Abortion
  - Gay rights
- The arguments and key points can be "pro" the topic, or "against" it

Examples

- Below are some examples of matching and non-matching pairs
- Matching
  - Argument: Prevents a large number of diseases
  - Key point: Vaccines prevent diseases
  - Matching because "Prevents a large number of diseases" is understable given the topic of "Vaccination should be mandatory"
- Non Matching
  - Argument: Vaccination should not be mandatory
  - Key Point: Vaccines have side-effects
  - Non matching because the key point mentions the aspect of side effect and the argument does not
- Non Matching
  - Argument: That is not possible.
  - Key point: Vaccinating everyone is not possible
  - Non matching because the argument is not an understandable sentence by itself, i.e. it is too vague.

E Computational Cost

All the experiments on method were done on V100-32gb. Total training and evaluation for the method took about 12 hours, with each run about 30 minutes.