HyEnA: A Hybrid Method for Extracting Arguments from Opinions

Paper ID: 2893

Abstract

Policy makers involve citizens in decision-making processes to harvest ideas and to build support for policies. Citizens' feedback should be analyzed quickly and accurately, and automated methods can assist in this process. Fully automated methods suffer from two main problems: (1) they require large labeled datasets, and (2) they work well for known viewpoints, but not for novel points of view.

We propose HyEnA, a hybrid (human + AI) method, to combine the speed of automated processing with the understanding and reasoning capabilities of humans. HyEnA extracts insights from textual feedback on policy options in the form of key arguments. We evaluate HyEnA rigorously on three feedback corpora on COVID-19 relaxation measures. We find that, on the one hand, a state-of-the-art automated method only covers about 50% of the arguments found by HyEnA and achieves less precision than HyEnA, justifying the need for human insight. On the other hand, HyEnA requires less human effort but does not compromise output quality compared to (fully manual) expert analysis, demonstrating the benefit of combining human and machine intelligence.

1 Introduction

To make decisions on large public issues, such as enacting measures to combat the COVID-19 pandemic and transitioning to green energy, policy makers often turn to the public for feedback (Lee, Hwang, and Moon 2020; Kythreotis et al. 2019). This feedback provides insights on the public opinion and contains diverse perspectives. Further, involving the public in the decision-making process helps in gaining their support when the decisions are to be implemented.

In the face of crises, decisions must be made swiftly. Thus, the collection of feedback, its analyses, and recommendations for decision making are done under tight time constraints. For example, when debating whether to relax COVID-19 measures in the Netherlands, researchers had a month to design the experiment, collect public feedback, and make recommendations (Mouter, Hernandez, and Itten 2021). The time constraint limits the amount of information researchers can look at, potentially painting an incomplete picture of the opinions. In the scenario above, researchers analyzed data manually, and they could analyze less than 8% of the feedback provided by more than 25,000 participants.

Argument Mining (AM) (Lawrence and Reed 2020) methods can assist in increasing the efficiency of feedback analysis by, e.g., separating strongly argumentative feedback from noise and classifying statements as supporting or opposing a decision. However, applying AM methods for feedback analysis poses three main challenges. First, AM methods generalize poorly across domains (Stab et al. 2018; Thorn Jakobsen, Barrett, and Søgaard 2021). Thus, they require large amounts of domain-specific training data, which is often not available. While contextualized representations, using the pre- or fine-tuning paradigm yield more promising results (Reimers et al. 2019b), they still rely on large amounts of data to be effective. Second, although AM methods can automatically detect logical connections between comments and policy decisions, they do not compress the information. That is, they do not recognize whether two identified arguments describe the same concept, leaving the policy makers with significant manual labor. Finally, analyzing a small sample of comments might cause minority opinions to be ignored (Klein 2012), creating a bias toward popular (repeated) arguments, which can perpetuate echo chambers and filter bubbles (Price 1989; Schulz-Hardt et al. 2000).

The *key point analysis* (KPA) task (Bar-Haim et al. 2020a) seeks to compress argumentative discourse into unique *key points*, which can be matched to arguments. However, synthesizing key points is a significant challenge. Bar-Haim et al. (2020a) employ domain experts (skilled debaters) to extract key points. However, such key points are not grounded in data (public opinion) and are subject to the perspectives and biases of the human experts. Further, making use of a few experts to generate key points defeats the purpose of engaging the public in the decision-making process.

We argue for a joint human-machine approach, taking advantage of the speed of automated methods and the human understanding of subtle issues. We propose HyEnA (Hybrid Extraction of Arguments), a hybrid (human + AI) method for extracting a diverse set of arguments from a textual corpus of opinions. HyEnA breaks down the argument extraction task into argument *annotation* and *consolidation* phases. In each phase, HyEnA employs human (crowd) annotators, but supports them via intelligent algorithms based on natural language processing (NLP). The HyEnA method

is is integrated in a web platform to support the humans in following the method end to end.

By employing humans, we take advantage of their semantic knowledge to interpret and condense user opinions into arguments. Further, since HyEnA requires the annotators to base their analysis on public feedback, the resulting arguments are grounded in data. Through intelligent sampling and merging techniques, and by resorting to overlapping annotations, we combine the judgements from individual annotators and incorporate a plurality of perspectives. Automated techniques augment the manual work to reduce the overall effort required for extracting arguments.

We evaluate our method on three corpora, each containing more than 10K public opinions on relaxing a COVID-19 restriction (Mouter, Hernandez, and Itten 2021). We compare HyEnA with an automated approach (Bar-Haim et al. 2020b), which generates key points from the corpus using a pretrained neural argument matching model. In addition, we compare the key arguments generated by HyEnA with insights identified by Mouter, Hernandez, and Itten (2021).

Contributions (1) We present a hybrid method for argument extraction, that, given a collection of opinionated user comments, generates a diverse set of key arguments that summarize the context under discussion. (2) We evaluate our method on real corpora of public feedback on policy options. Compared to an automated baseline, HyEnA increases both precision and coverage of the key arguments produced. Compared to the manual baseline, HyEnA identifies a large portion of arguments identified by experts as well as new arguments that experts did not identify.

2 Related work

We describe closely related works on Argument Mining, their application to opinion analysis, and methods that aim to extract key arguments from an opinion corpus.

2.1 Argument Mining

Argument Mining (AM) methods (Lawrence and Reed 2020; Cabrio and Villata 2018) focus on computational analysis of arguments. They seek to discover arguments brought forward by speakers and identify connections between them. AM is a costly and complex process, and it often requires significant effort by human annotators for reaching moderate inter-rater agreement (Teruel et al. 2018).

The ability to recognize and extract arguments from text is dependent on the argumentativeness of the underlying data. Given argumentative texts, popular NLP models are reasonably good at recognizing argumentative discourse (Niculae, Park, and Cardie 2017; Eger, Daxenberger, and Gurevych 2017; Reimers et al. 2019b). Typically, the first step of AM is to identify the elemental components of arguments (e.g., *claims* ans *premises*) in text (Toulmin 2003). The combination of such components forms a structured argument. However, there is currently no consensus on the exact nature of such elemental components (Daxenberger et al. 2017). Nonetheless, a few characteristics have been recognised as important for recognizing arguments, namely that arguments (1) contain logical reasoning (Stab and Gurevych 2014), (2) address a *why* question (Biran and Rambow 2011), and (3) have a non-neutral stance towards the issue being discussed (Stab and Gurevych 2014).

HyEnA is a novel AM method that employs human annotators alongside automated NLP models. By splitting up the argument extraction task into distinct phases, we hope to take advantage of the diverse perspectives from humans, while addressing scalability problems through automation. Because annotators are only given the opinion text, we aim to achieve better grounding by preserving links between argument and the original text, all while providing condensed key arguments useful in analysis.

2.2 Summarization of Arguments in Discussions

Automated methods have been proposed to create a core set of key points from a large list of individual arguments (Bar-Haim et al. 2020b). In this paradigm, comments are filtered by a manually tuned selection heuristic, resulting in a list of key point candidates (Gretz et al. 2020). The candidates are matched against all comments, based on a classifier trained for the argument–key point matching task (Bar-Haim et al. 2020a). Such an automated key point extraction method has been shown to perform well on the same domain, and be applicable in cross-domain settings.

We employ a state-of-the-art key point extraction method (Bar-Haim et al. 2020a) as an automated baseline to compare HyEnA against. We evaluate the performance of these approaches on a novel domain on COVID-19 measures.

Finally, there exists a body of work on semantic textual similarity (STS) and Natural Language Inference (NLI). In these works, models are trained to indicate semantic similarity or logical entailment between two sentences (Reimers et al. 2019a; Conneau et al. 2017). They have made significant impact for general purpose applications (Xu, He, and Li 2018; Zhong et al. 2020). However, for downstream application, they often need additional fine-tuning (Howard and Ruder 2018) in order to perform a task well. They also capture generic aspects of semantic similarity and entailment, which may not be applicable to arguments (Reimers et al. 2019a), or conversely overfit to spurious patterns in the data (McCoy, Pavlick, and Linzen 2019).

3 Method

HyEnA is a hybrid method since it combines automated techniques and human judgement (Akata et al. 2020). HyEnA guides human annotators toward the creation of a list of *key arguments* (i.e., a list of semantically distinct arguments that describe relevant aspects of the topic under discussion) from an *opinion corpus* composed of individual *opinions* (i.e., textual comments) on the topic of discussion.

HyEnA consists of two phases as depicted in Figure 1. In the first phase (*Key Argument Annotation*), an intelligent sampling algorithm guides human annotators through an opinion corpus to extract high-level information from the opinions. In the second phase (*Key Argument Consolidation*), a new group of annotators merges the results from the first phase, supported by an intelligent merging strategy, involving manual and automatic labeling. Through this second phase, HyEnA aims to reduce the effect of subjectivity

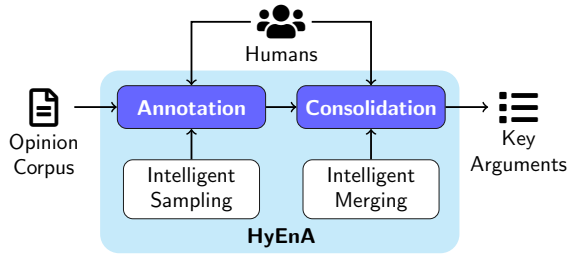

Figure 1: Setup of the HyEnA method.

of annotation. The final result of HyEnA is a list of key arguments grounded on the opinions in the corpus.

3.1 Opinion Corpora

Our opinion corpus is composed of citizens' feedback on COVID-19 relaxation measures, a contemporary topic. The feedback was gathered during April and May 2020 using the Participatory Value Evaluation (PVE) method (Mouter, Hernandez, and Itten 2021). In the PVE, participants are offered a set of policy options and asked to select their preferred portfolio of choices. Then, the participants are asked to motivate why they picked certain options (*pro* stance) and not pick the other options (*con* stance) via textual comments. Pro- and con-opinions together form the opinion corpus.

The PVE collected feedback from 26,293 Dutch citizens on eight policy options about COVID-19 relaxation measures. We analyze the feedback on three of these options, treating feedback on each option as an opinion corpus. Table 1 shows the policy options, an example opinion for each option, and the number of opinions per option. For each policy option, we use the keyword in uppercase as the policy (or corpus) identifier in the remainder of the paper.

Table 1: Examples of opinions in the COVID-19 corpora.

Policy option (Corpus)	**Example opinion**	**# Opinions**
YOUNG people may come together in small groups	Then they can go back to school (Pro)	13,400
All restrictions are lifted for persons who are IMMUNE	Encourages inequality (Con)	10,567
REOPEN hospitality and entertainment industry	The economic damage is too high (Pro)	12,814

The opinions in these corpora are similar to noisy user-generated web comments, as in Habernal and Gurevych (2017). Some opinions span multiple sentences and contain more than one argument. In our experiments, the HyEnA method is applied to one corpus at a time.

The original opinions were provided in Dutch. To accommodate a diverse set of annotators in our experiments, we translated all comments to English using the Microsoft Azure Translation[1] service. All experiments are performed with the translated opinions. Mixing (pretrained) embeddings and machine-translated comments has a minimal impact on downstream task performance (Sennrich, Haddow, and Birch 2016; Eger et al. 2018; Daza and Frank 2020). Although all experiments are conducted in English, the link to the original Dutch text is preserved for future applications.

[1]https://azure.microsoft.com/en-us/services/cognitive-services/translator/

3.2 Key Argument Annotation

In the first phase of HyEnA, human annotators extract individual key argument lists by analysing the opinion corpus. Since a realistic corpus may consist of thousands of opinions, it is unfeasible for an annotator to read all opinions. Thus, HyEnA proposes a fixed number of opinions to each annotator. To maximize the coverage and diversity of the opinions, HyEnA employs NLP and a sampling technique to select the opinions presented to each annotator.

Opinion Selection Each annotator is presented, one at a time, a fixed number of opinions. To select the next opinion to present to an annotator, first, we embed all opinion and arguments annotated thus far using the S-BERT model (M_{SBERT}) (Reimers et al. 2019a). Then, we select a pool of candidate opinions using the Farthest-First Traversal (FFT) algorithm (Basu, Banerjee, and Mooney 2004).

FFT selects the candidate pool as the f farthest opinions in the embedding space from the previously read opinions and annotated arguments (in our experiments, we empirically selected $f = 5$). Next, we use an argument quality classifier trained on Gretz et al. (2020) to select the opinion most clear and connected to the proposal among the five farthest. In this way, we aim at increasing the diversity and quality of the opinions presented to each annotator.

Annotation Upon reading an opinion, the annotator is asked, first, to *identify* whether the opinion contains an argument or not. If so, the annotator is asked to check whether the argument is already included in their current list of key arguments. If not, the annotator should *extract* the argument into a standalone expression (i.e., into a key argument), and add it to the list of key arguments. When adding a new argument, the annotator is asked to indicate the *stance* of the opinion (i.e., whether it is in support or against the related policy option). To facilitate this task, HyEnA highlights the most probable stance for the user as a label suggestion (Schulz et al. 2019; Beck et al. 2021).

Topic Assignment We assign each key argument created by an annotator to the topics generated with a BERTopic model, $\mathcal{T}$, trained on all the available opinions (Grootendorst 2020). We create a short-list of topics, selected as the most frequent topics found by $\mathcal{T}$, with duplicates and unintelligible topics manually removed by two experts. Per argument, we ask human annotators to select the associated topics from the generated short-list. The topics assignments are used in the second phase to compute argument similarity.

The output of the first phase are multiple key argument lists (one per annotator), each containing key arguments and their stances. Further, each argument is manually assigned a topic mixture over a pre-selected set of topics.

3.3 Key Argument Consolidation

In the first phase, (1) the annotators are exposed to a small subset of the opinions in the corpus, and (2) the interpretation of arguments is subjective. In the second phase, HyEnA seeks to *consolidate* the key argument lists generated in the first phase. Our goal is to increase the diversity of the resulting arguments and compensate for individual biases.

First, we create the union of all lists of key arguments generated in the first phase of HyEnA. Then, we ask the annotators to evaluate the similarity of the key argument pairs in the union list. Based on the similarity labels, we employ a clustering algorithm to group similar key arguments, producing a consolidated list of key arguments.

Pairwise Annotation To simplify the consolidation task, we present one pair of key arguments, at the time, to the annotators and ask them whether the concepts described by the two key arguments in the pair are semantically similar. To reduce annotation effort, we ask the annotators to annotate only the most informative key argument pairs, and automatically annotate the remaining pairs. To select the most informative pairs, we adapt the Partial-Ordering approach, POWER (Chai et al. 2016), as described below.

Let p_{ij} be a pair of key arguments $\langle a_i, a_j \rangle$. The similarity between the two key arguments in the pair is described by a set of *similarity scores*, s^h_{ij}, which indicate how similar the key arguments are. By using multiple scores, we seek to make the similarity computation robust. For each p_{ij}, we compute two similarity scores described in Table 2. We use cosine similarity for s^1_{ij} since the angular distance describes the semantic similarity between two arguments. In contrast, we use Euclidean distance for s^2_{ij} since the absolute values of the topic assignment are relevant.

Table 2: The two similarity scores between key argument pairs used to create the pairwise dependency graph.

Measure	Description
$s^1_{ij} = \frac{\mathbf{i} \cdot \mathbf{j}}{\|\mathbf{i}\| \|\mathbf{j}\|}$	Cosine similarity between embeddings $\mathbf{i} = M_{SBERT}(a_i)$ and $\mathbf{j} = M_{SBERT}(a_j)$
$s^2_{ij} = \frac{1}{d(T(a_i), T(a_j))}$	Inverse of the Euclidean distance d between manual topic assignments T of a_i and a_j

Given the similarity scores, we construct a dependency graph G (as in the top-left part of Figure 2), where each key argument pair is a node in G and the edges indicate a Pareto dependency ($\succ$) between two pairs as follows:

$$p_{ij} \succeq p_{i'j'} \quad \text{if} \quad \forall h \quad s^h_{ij} \geq s^h_{i'j'} \tag{1}$$

$$p_{ij} \succ p_{i'j'} \quad \text{if} \quad p_{ij} \succeq p_{i'j'} \quad \text{and} \quad \exists h \quad s^h_{ij} > s^h_{i'j'} \tag{2}$$

Next, we follow POWER to extract disjoint paths from G. The highlighted path in the bottom-left part of Figure 2 is an example disjoint path. For every path, we perform a pairwise annotation as in the right part of Figure 2. We select the vertex at the middle of the unlabeled portion of the path, and ask multiple humans to indicate whether the concepts described by the two arguments in the pair are similar, and select the annotation with majority agreement. Given the annotation,

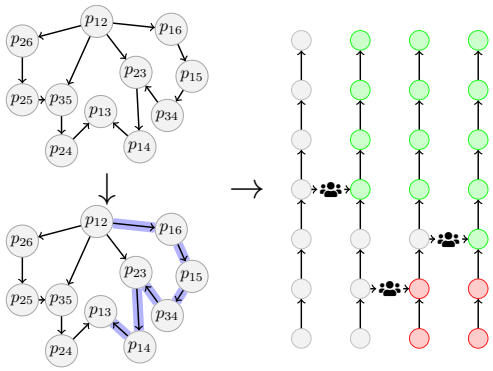

Figure 2: Pairwise annotation from dependency graph.

we can automatically label (1) all following pairs in the path as similar in case the vertex is labeled as similar, or (2) all preceding pairs in the path as non-similar in case the vertex is labeled as non-similar. In essence, using the Pareto dependency, we search for threshold similarity scores for each path, above which all pairs are considered similar, and below which all pairs are non-similar. Because this is a local threshold, we prevent over-generalization.

Ultimately, through the combination of manual and automatic labeling, we obtain a similarity label for all possible key argument pairs. To annotate the complete graph efficiently, we employ the parallel Multi-Path annotation algorithm described in the Appendix.

Clustering Given a similarity label for each key argument pair, our goal is to identify a consolidated list of unique key arguments. However, the similarity among key arguments may not be transitive. That is, given $\langle a_1, a_2 \rangle$ as similar and $\langle a_2, a_3 \rangle$ as similar, $\langle a_1, a_3 \rangle$ may be labeled as dissimilar. This can be happen because (1) the interpretation of similarity can be subjective (for the manually labeled pairs), and (2) the automatic approach is not always accurate (for the automatically labeled pairs). Thus, we employ a clustering algorithm for identifying a consolidated list. First, we construct a similarity graph, where each key argument is a node and there is an edge between two arguments, if they are labeled as similar. Then, we employ a graph clustering algorithm for recognizing argument clusters. We experiment with Louvain and spectral clustering (Section 4.2).

4 Experimental Setup

We execute and evaluate HyEnA, involving 327 Prolific (www.prolific.co) crowd workers as annotators. We required the workers to be fluent in English, have an approval rate above 95%, and completed at least 100 submissions. Our experiment was approved by an Ethics Committee and we received an informed consent from each subject.

Table 3 shows an overview of the tasks involved in the experiment. First, we ask the annotators to perform the HyEnA method to generate lists of key arguments for three corpora (Section 3.1). Then, we evaluate the quality of the obtained lists of key arguments in comparison with lists generated for the same corpora via two baselines. All tasks except topic

generation were performed by the crowd workers. The detailed instructions provided to the annotators in each task are included as supplemental material.

Table 3: Experiment overview. Items to be annotated can be opinions (O), arguments (A), topics (T), or combinations.

Task	Policy Option	# Items	# Annotators
Key argument annotation	YOUNG	255 (O)	5
	IMMUNE	255 (O)	5
	REOPEN	255 (O)	5
Topic generation	all	45 (T)	2
Topic assignment	YOUNG	90 (A)	10
	IMMUNE	64 (A)	5
	REOPEN	69 (A)	5
Key argument consolidation	YOUNG	1538 (A+A)	99
	IMMUNE	824 (A+A)	57
	REOPEN	940 (A+A)	87
Key argument evaluation	YOUNG	169 (O+A)	21
	IMMUNE	112 (O+A)	14
	REOPEN	145 (O+A)	14

4.1 Phase 1: Key Argument Annotation

In the first phase of HyEnA, each annotator extracts a key arguments list from an opinion corpus. In each corpus, five annotators annotated 51 opinions each, for a total of 255 opinions. Of the 51 opinions, the first is selected randomly, and the following 50 are selected by FFT. This number of opinions was empirically selected to make the annotation feasible within a maximum of one hour.

Topics We train a BERTopic model on each of opinion corpus. The model generated 59, 56, and 72 topics for the YOUNG, IMMUNE, and REOPEN corpus, respectively. Since the number of topics was very high for manual assignment of arguments to topics, we curate a short-list of topics per corpus. To do so, we select 15 most frequent topics in a corpus and ask two experts to remove duplicates (i.e., topics covering the same semantic aspect) and rate the clarity (i.e., how well the topic describes a relevant aspect of the discussion in the corpus) of each topic. The topics with an average clarity scores above 2.5 composed the short-list of topics. Then, we asked the annotators to assign topics to each key argument generated in the first phase of HyEnA.

4.2 Phase 2: Key Argument Consolidation

In the second phase of HyEnA, we obtain similarity labels $y(a_i, a_j)$ (1 if similar, 0 if not) for all key argument pairs $\langle a_i, a_j \rangle$—some pairs are labeled by the annotators and others are automatically labeled. Given the similarity labels, we construct an argument similarity graph, and cluster the graph to identify a consolidated list of key arguments.

Key Argument Clustering We experiment with two well-known graph clustering algorithms:

- Louvain clustering (Blondel et al. 2008) uses network modularity to identify groups of vertices to cluster. It includes a resolution parameter, r.
- Self-tuning spectral clustering (Zelnik-Manor and Perona 2004) uses dimensionality reduction in combination with k-means to obtain clusters. It includes the desired number of clusters, k, as parameter.

We choose the parameters of these algorithms to minimize the error metric E shown in Equation 3. The metric is calculated over all obtained clusters K, and it penalizes clusters having dissimilar argument pairs. That is, for a cluster k, $\forall a_i, a_j \in k$, if $y(a_i, a_j) = 1$, then the error for that cluster is 0. If a cluster contains only a single element, we manually set the error for that cluster to 1, to discourage creating too many single-member clusters.

$$E = \frac{1}{|K|} \sum_{k \in K} \frac{\sum_{a_i, a_j \in k} \mathbb{1}_{y(a_i, a_j)=0}}{\binom{|k|}{2}} \tag{3}$$

4.3 Baseline Comparison

We compare HyEnA to automated and manual baselines.

Automated Baseline We use the **ArgKP** argument matching model (Bar-Haim et al. 2020b) to automatically extract key points from the corpus. ArgKP selects candidate key points from opinions using a manually-tuned heuristic, which filters opinions on their lengths and form. Following Bar-Haim et al. (2020b), we adjusted ArgKP parameters such that 20% of the opinions are selected as candidates by the heuristic. Opinions are provided a match score using a pretrained matching network based on RoBERTa (Liu et al. 2019). Opinions only match the highest scoring candidate key points if their match score exceeds a threshold θ (corresponding to the BM+TH approach). After deduplication, this results in a single list of key arguments per corpus.

To compare HyEnA and the automated baseline, we adopt the following approach similar to Bar-Haim et al. (2020b).

1. Run HyEnA and extract the set of key arguments A_1 from the opinions O_1^{obs} observed by annotators. Each argument $a_i \in A_1$ is extracted from an opinion $o_i \in O_1$, where $O_1 \subseteq O_1^{obs}$ is the set of opinions annotated with a key argument during the first phase of HyEnA.
2. Run the ArgKP model on the corpus and extract a set of key arguments A_2 based on the entire set of opinions O_2.
3. Sample pairs of corresponding opinions and arguments, (o_i, a_i), where $o_i \in O_1 \cap O_2$ and $a_i \in A_1 \cup A_2$.
4. Ask annotators to label $z(o_i, a_i) = 1$ for all matching pairs and $z(o_i, a_i) = 0$ for all non-matching pairs, and keep the majority consensus from multiple annotators.

We compute two metrics to evaluate the quality of the extracted key arguments. The *coverage* (C) indicates the capability of a method to extract a diverse set of arguments. Since O_1^{obs} consists of a variety of opinions, the ability to synthesize key arguments from these opinions is important.

$$C = \frac{|A_1|}{|O_1^{obs}|} \tag{4}$$

The *precision* (P) is the extent to which the extracted key arguments can be matched to the opinions in a corpus.

$$P = \frac{\sum_{o_i, a_i} \mathbb{1}_{z(o_i,a_i)=1}}{|A_1|} \quad (5)$$

Manual Baseline Mouter, Hernandez, and Itten (2021) involve six experts to manually analyze the feedback from a sample of participants (2,237 out of 26,293) over all eight policy options and identify key arguments. However, they do not report the exact number of opinions analyzed. Since there are 36,781 opinions for the three options we analyze (Table 1), we estimate the number of opinions the six experts would have analyzed to be 3,129 across the three options. In contrast, HyEnA annotators analyze 765 intelligently selected opinions across the three options.

HyEnA reduces the number of opinions analyzed. Then, an important question is the extent to which the key argument lists generated by HyEnA and the manual baseline have comparable insights. To answer this question, we compute the number of HyEnA key arguments that are overlapping, missing, and new compared to the expert-identified key arguments. We could not compute precision and coverage for the manual baseline because it does not include a mapping between key arguments and opinions.

5 Results and Discussion

First, we show the influence of the intelligent sampling and merging techniques HyEnA employs in Phases 1 and 2. Next, we compare HyEnA with the automated and manual baselines to evaluate the benefits of HyEnA's hybrid approach. Finally, we analyze the inter-rater reliability.

5.1 Phase 1: Key Argument Annotation

Table 4 show the number of different Phase 1 operations annotators perform. On average, the annotators identified 15 unique key arguments per option. Roughly half of the opinions were skipped. The main reason for skipping was an opinion not having a clear argument for supporting or opposing an option. This is a positive result since the noise (irrelevant or non-argumentative opinions) in public feedback can be much higher. We conjecture that the argument quality classifier we incorporate for opinion selection is effective in filtering the noise. Further, the annotators marked only about 15% of the encountered opinions as already annotated key arguments, which shows that the FFT approach is effective in sampling a diverse set of opinions for annotation.

Table 4: Summary of Phase 1 results, showing the average numbers (and standard deviation) of annotation operations.

Option	# Args	# Skip	# Already Annotated
YOUNG	18.0 (5.5)	23.4 (5.4)	11.4 (9.0)
IMMUNE	12.8 (2.6)	31.4 (4.5)	8.6 (4.4)
REOPEN	13.8 (7.6)	29.2 (11.5)	10.2 (7.6)

5.2 Phase 2: Key Argument Consolidation

Table 5 shows the benefit POWER, HyEnA's approach for merging key argument lists. On average, POWER reduces the number of pairs requiring human annotation by 60%.

Table 5: Summary of Phase 2 annotations. # Reduced is the reduced number of pairs (from # Pairs) requiring a manual label. HyEnA consolidated A_1 arguments into K clusters.

Option	# Pairs	# Reduced Pairs	τ	$\lvert A_1 \rvert$	$\lvert K \rvert$
YOUNG	4005	1538	0.34	90	20
IMMUNE	2016	824	0.42	64	14
REOPEN	2346	943	0.41	69	18

The transitivity score τ (Newman, Watts, and Strogatz 2002) indicates the extent to which transitivity holds among the similarity labels of argument pairs. The low transitivity scores justify the clustering we perform. As Figure 3 shows, with the best parameter setting, Louvain clustering yields smallest error for YOUNG and IMMUNE corpora, and spectral clustering yields the smallest error for REOPEN corpus.

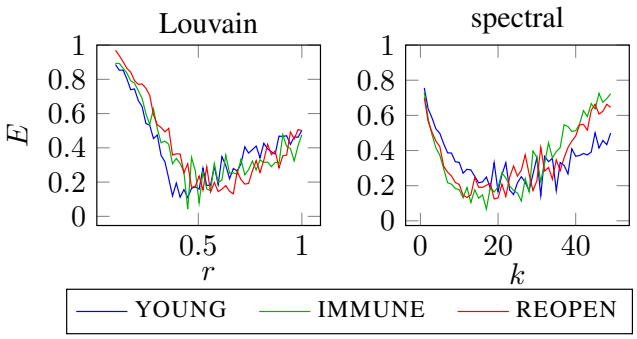

Figure 3: Parameter tuning for key argument clustering.

5.3 Comparison with Automated Baseline

Figure 4 compares the coverage and precision of HyEnA and ArgKP. The coverage (portion of opinions that are annotated with a key argument) is low for both methods. This is because several opinions are annotated as not containing argumentative content. Yet, the coverage of HyEnA (0.34 on average) is higher than the coverage of ArgKP (0.11 on average). ArgKP often failed to recognize the key arguments in the diverse set of opinions included by HyEnA.

ArgKP yields a larger number of key arguments (between 30 and 40 for each option) than HyEnA. However, these arguments lead to an average precision of 0.37. In contrast, HyEnA extracted less argument clusters, but with a higher precision (0.80). Further, we notice that human annotators actively rephrase the content of the key arguments—a significant part of the argument text (more than half) is directly copied from the opinion text only in 22% of the annotations.

5.4 Comparison with Manual Baseline

Table 6 shows a confusion matrix, comparing overlapping (yes, yes), missing (no, yes), and new (yes, no) key arguments between HyEnA and the manual baseline. Recall that HyEnA required an analysis of 765 opinions, whereas the

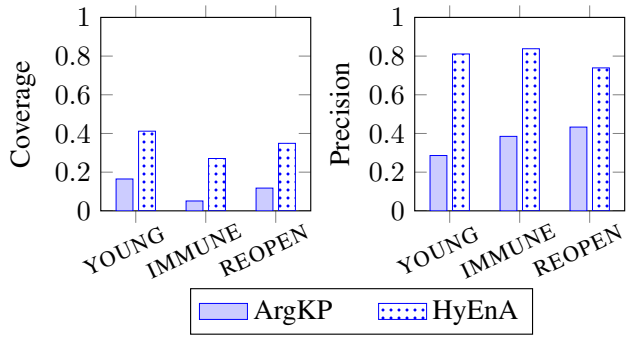

Figure 4: Coverage and Precision for HyEnA and ArgKP.

manual baseline required 3,129 opinions, to produce their respective key arguments lists. Despite incurring less human effort, HyEnA list has a large overlap with the expert list.

Table 6: Confusion matrix, comparing the key argument lists of HyEnA and manual baseline.

		Manual baseline					
		YOUNG		IMMUNE		REOPEN	
		yes	no	yes	no	yes	no
HyEnA	yes	8	7	7	2	10	1
	no	1	–	0	–	4	–

HyEnA missed some key arguments that experts identified. For instance, a key argument about building up herd immunity was not in the HyEnA list for the REOPEN option. Interestingly, ArgKP also failed to find such arguments. We conjecture that increasing the number of opinions annotated in HyEnA would subsequently yield the missing insights.

Finally, HyEnA also led to new insights that were missed by experts. For instance, the argument about the physical well-being of young people, which seems important, was not in the expert list for the YOUNG option. This is potentially because the random sample analyzed by the experts did not include opinions supporting this argument. In contrast, the diverse set of opinions intelligently sampled by HyEnA included opinions supporting this argument.

5.5 Annotator Reliability

Table 7 shows the inter-rater reliability (IRR) for four steps with overlapping human annotations. In the topic generation phase (Section 4.1), we use the intraclass correlation coefficient ICC$(3, k)$ (Shrout and Fleiss 1979) since it involves ordinal ratings. In the other three tasks, multiple binary labels are obtained for the same subjects. In these tasks, we use prevalence- and bias-adjusted κ (PABAK) (Sim and Wright 2005), which adjusts Fleiss' κ for prevalence and bias resulting from small or skewed distribution of ratings.

The IRR for topic generation and assignment tasks are substantial. The IRR for key argument consolidation and baseline comparison are fair and moderate, respectively.

Table 7: Average (and standard deviation) IRR scores.

Task	**ICC3k**	**PABAK**
Topic generation	0.66 (0.14)	–
Topic assignment	–	0.81 (0.10)
Key argument consolidation	–	0.34 (0.03)
Key argument evaluation	–	0.43 (0.08)

The relatively low IRR scores of consolidation and baseline comparison tasks are not shortcomings of the HyEnA method in itself. Instead, they demonstrate the inherent difficulty and subjectivity involved in interpreting the arguments, and matching the arguments and opinions. This also justifies the need for a robust argument consolidation phase that integrates judgements from a range of interpretations.

6 Conclusion and Directions

We develop and evaluate HyEnA, a hybrid method that combines human judgements with automated methods to generate a diverse set of key arguments. We show that HyEnA extracts key arguments from a large corpus of noisy opinions with higher precision and coverage than a state-of-the-art automated method for key point analysis. Additionally, HyEnA provides important insights that were not included in an expert-driven analysis over the same corpus despite requiring an analysis of a less number of opinions.

The pairwise comparison task of the consolidation phase is the most human resource intensive task in HyEnA. Also, comparing arguments is cognitively demanding for annotators. If we increase the number of opinions analyzed in the first phase to increase the coverage of HyEnA, the consolidation effort in the second phase will increase further. We employed an automated technique, which reduced the number of comparisons required in the consolidation phase by 60%. Additional research is necessary to further reduce the consolidation effort. For example, first clustering the key arguments and then consolidating the arguments within these clusters (reverse order as HyEnA) can influence the performance and the effort, but requires further investigation.

Annotators only reach a fair and moderate agreements in the consolidation and argument matching tasks. This shows the complexity of language understanding, and the subtleties involved in interpreting and reasoning about arguments and opinions. We pose that hybrid approaches which use human insight are a key component for public feedback analysis. Uncovering these subtleties and making them explicit is a crucial task for enabling effective perspective taking (Chen et al. 2019). Finding arguments from a large discourse is only one of the aspects that constitute the perspectives in a discussion. Future work can incorporate analysis over the same discourse for values (Liscio et al. 2021) or other perspective factors, such as sentiment, emotion, and attribution (van Son et al. 2016). By combining these rich aspects with arguments, we can merge the logical basis of the discussion with other semantic and syntactic information, possibly allowing a close scrutiny of the perspectives at play.

References

Akata, Z.; Balliet, D.; De Rijke, M.; Dignum, F.; Dignum, V.; Eiben, G.; Fokkens, A.; Grossi, D.; Hindriks, K.; Hoos, H.; et al. 2020. A Research Agenda for Hybrid Intelligence: Augmenting Human Intellect With Collaborative, Adaptive, Responsible, and Explainable Artificial Intelligence. *Computer*, 53(8): 18–28.

Bar-Haim, R.; Eden, L.; Friedman, R.; Kantor, Y.; Lahav, D.; and Slonim, N. 2020a. From Arguments to Key Points: Towards Automatic Argument Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4029–4039.

Bar-Haim, R.; Kantor, Y.; Eden, L.; Friedman, R.; Lahav, D.; and Slonim, N. 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)*, 39–49.

Basu, S.; Banerjee, A.; and Mooney, R. J. 2004. Active Semi-Supervision for Pairwise Constrained Clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, SDM '04, 333–344. Orlando, Florida, USA: Society for Industrial and Applied Mathematics.

Beck, T.; Lee, J.-U.; Viehmann, C.; Maurer, M.; Quiring, O.; and Gurevych, I. 2021. Investigating label suggestions for opinion mining in German Covid-19 social media. 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)*, 1–13. Online: Association for Computational Linguistics.

Biran, O.; and Rambow, O. 2011. Identifying Justifications in Written Dialogs. In *2011 IEEE Fifth International Conference on Semantic Computing*, 162–168. IEEE.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10): P10008.

Cabrio, E.; and Villata, S. 2018. Five Years of Argument Mining: a Data-driven Analysis. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 5427–5433. International Joint Conferences on Artificial Intelligence Organization.

Chai, C.; Li, G.; Li, J.; Deng, D.; and Feng, J. 2016. Cost-effective crowdsourced entity resolution: A partial-order approach. In *Proceedings of the 2016 International Conference on Management of Data*, 969–984.

Chen, S.; Khashabi, D.; Yin, W.; Callison-Burch, C.; and Roth, D. 2019. Seeing Things from a Different Angle: Discovering Diverse Perspectives about Claims. 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)*, 542–557.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 670–680.

Daxenberger, J.; Eger, S.; Habernal, I.; Stab, C.; and Gurevych, I. 2017. What is the Essence of a Claim? Cross-Domain Claim Identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2055–2066.

Daza, A.; and Frank, A. 2020. X-SRL: A Parallel Cross-Lingual Semantic Role Labeling Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3904–3914.

Eger, S.; Daxenberger, J.; and Gurevych, I. 2017. Neural End-to-End Learning for Computational Argumentation Mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11–22.

Eger, S.; Daxenberger, J.; Stab, C.; and Gurevych, I. 2018. Cross-lingual Argumentation Mining: Machine Translation (and a bit of Projection) is All You Need! In *Proceedings of the 27th International Conference on Computational Linguistics*, 831–844. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Gretz, S.; Friedman, R.; Cohen-Karlik, E.; Toledo, A.; Lahav, D.; Aharonov, R.; and Slonim, N. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7805–7813.

Grootendorst, M. 2020. BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics. https://maartengr.github.io/BERTopic/.

Habernal, I.; and Gurevych, I. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1): 125–179.

Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339.

Klein, M. 2012. Enabling large-scale deliberation using attention-mediation metrics. *Computer Supported Cooperative Work (CSCW)*, 21(4-5): 449–473.

Kythreotis, A. P.; Mantyka-Pringle, C.; Mercer, T. G.; Whitmarsh, L. E.; Corner, A.; Paavola, J.; Chambers, C.; Miller, B. A.; and Castree, N. 2019. Citizen social science for more integrative and effective climate action: A science-policy perspective. *Frontiers in Environmental Science*, 7: 10.

Lawrence, J.; and Reed, C. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4): 765–818.

Lee, S.; Hwang, C.; and Moon, M. J. 2020. Policy learning and crisis policy-making: quadruple-loop learning and COVID-19 responses in South Korea. *Policy and Society*, 39(3): 363–381.

Liscio, E.; van der Meer, M.; Siebert, L. C.; Jonker, C. M.; Mouter, N.; and Murukannaiah, P. K. 2021. Axies: Identifying and Evaluating Context-Specific Values. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 799–808. London.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

McCoy, T.; Pavlick, E.; and Linzen, T. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3428–3448.

Mouter, N.; Hernandez, J. I.; and Itten, A. V. 2021. Public participation in crisis policymaking. How 30,000 Dutch citizens advised their government on relaxing COVID-19 lockdown measures. *Plos one*, 16(5): e0250614.

Newman, M. E.; Watts, D. J.; and Strogatz, S. H. 2002. Random graph models of social networks. *Proceedings of the national academy of sciences*, 99(suppl 1): 2566–2572.

Niculae, V.; Park, J.; and Cardie, C. 2017. Argument Mining with Structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 985–995.

Price, V. 1989. Social identification and public opinion: Effects of communicating group conflict. *Public Opinion Quarterly*, 53(2): 197–224.

Reimers, N.; Gurevych, I.; Reimers, N.; Gurevych, I.; Thakur, N.; Reimers, N.; Daxenberger, J.; and Gurevych, I. 2019a. 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.

Reimers, N.; Schiller, B.; Beck, T.; Daxenberger, J.; Stab, C.; and Gurevych, I. 2019b. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 567–578.

Schulz, C.; Meyer, C. M.; Kiesewetter, J.; Sailer, M.; Bauer, E.; Fischer, M. R.; Fischer, F.; and Gurevych, I. 2019. Analysis of Automatic Annotation Suggestions for Hard Discourse-Level Tasks in Expert Domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2761–2772.

Schulz-Hardt, S.; Frey, D.; Lüthgens, C.; and Moscovici, S. 2000. Biased information search in group decision making. *Journal of personality and social psychology*, 78(4): 655.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 86–96. Berlin, Germany: Association for Computational Linguistics.

Shrout, P. E.; and Fleiss, J. L. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2): 420.

Sim, J.; and Wright, C. C. 2005. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3): 257–268.

Stab, C.; and Gurevych, I. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, 1501–1510.

Stab, C.; Miller, T.; Schiller, B.; Rai, P.; and Gurevych, I. 2018. Cross-topic Argument Mining from Heterogeneous Sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3664–3674.

Teruel, M.; Cardellino, C.; Cardellino, F.; Alemany, L. A.; and Villata, S. 2018. Increasing argument annotation reproducibility by using inter-annotator agreement to improve guidelines. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Thorn Jakobsen, T. S.; Barrett, M.; and Søgaard, A. 2021. Spurious Correlations in Cross-Topic Argument Mining. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, 263–277. Online: Association for Computational Linguistics.

Toulmin, S. E. 2003. *The uses of argument*. Cambridge university press.

van Son, C.; Caselli, T.; Fokkens, A.; Maks, I.; Morante, R.; Aroyo, L.; and Vossen, P. 2016. GRaSP: A Multilayered Annotation Scheme for Perspectives. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 1177–1184.

Xu, J.; He, X.; and Li, H. 2018. Deep learning for matching in search and recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1365–1368.

Zelnik-Manor, L.; and Perona, P. 2004. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems*, 17: 1601–1608.

Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; and Huang, X.-J. 2020. Extractive Summarization as Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6197–6208.