From Arguments to Key Points: Towards Automatic Argument Summarization

Roy Bar-Haim   **Lilach Eden**   **Roni Friedman**   **Yoav Kantor**
Dan Lahav   **Noam Slonim**[*]
IBM Research
{roybar,lilache,roni.friedman-melamed,yoavka,noams}@il.ibm.com
dan.lahav@ibm.com

Abstract

Generating a concise summary from a large collection of arguments on a given topic is an intriguing yet understudied problem. We propose to represent such summaries as a small set of talking points, termed *key points*, each scored according to its salience. We show, by analyzing a large dataset of crowd-contributed arguments, that a small number of key points per topic is typically sufficient for covering the vast majority of the arguments. Furthermore, we found that a domain expert can often predict these key points in advance. We study the task of argument-to-key point mapping, and introduce a novel large-scale dataset for this task. We report empirical results for an extensive set of experiments with this dataset, showing promising performance.

1 Introduction

Governments, businesses and individuals, all need to make decisions on a daily basis: *"Should cannabis be legalized?"*, *"Should we develop this product?"*, *"Should I become a vegetarian?"*. When making an important decision, the process typically comprises several steps: first, we gather as much information as we can about the pros and cons of the proposal under consideration. We may then summarize the collected information as a short list of the main arguments for each side. Lastly, we aim to weigh the pro and con arguments against each other to make the final decision.

Where can we find relevant arguments for a given topic? In recent years, significant progress was made in the field of *argument mining*, automatic identification and extraction of argumentative structures in text (Lawrence and Reed, 2020). Specifically, several works focused on topic-related argument mining from the Web or other massive corpora (Levy et al., 2017, 2018; Wachsmuth et al.,

2017; Stab et al., 2018a,b; Ein-Dor et al., 2020). Policy makers in governments or businesses may also conduct surveys to collect from large audiences arguments supporting or contesting some proposal.

Each of the above methods may result in hundreds or thousands of arguments per topic, making it impossible for the decision maker to read and digest such large amounts of information. Several works aimed to alleviate this problem by clustering together related arguments, based on different notions of relatedness, such as *similarity* (Reimers et al., 2019), *frames* (Ajjour et al., 2019), and *argument facets* (Misra et al., 2016). These works, however, did not attempt to create a concise textual summary from the resulting clusters.

In this work we propose to summarize the arguments supporting each side of the debate by mapping them to a short list of talking points, termed *key points*. The salience of each key point can be represented by the number of its matching arguments. An example for such summary is shown in Table 1. Key points may be viewed as high-level arguments. They should be general enough to match a significant portion of the arguments, yet informative enough to make a useful summary.

The proposed method raises a fundamental question: can a small number of key points effectively summarize massive amount of arguments collected from a large population? In this work we give a positive answer to this question, based on extensive analysis over 28 controversial topics and 7,000 crowd-contributed pro and con arguments for these topics. Furthermore, we found that, given a controversial topic, a domain expert can compose a short, comprehensive list of key points even without looking at the arguments themselves.

Motivated by the above findings, we assume in this work that the key points for each topic are given, and focus on the task of automatically map-

---
[*] All authors equally contributed to this work.

Homeschooling should be banned	#Args
Pro	
Mainstream schools are essential to develop social skills.	61
Parents are not qualified as teachers.	20
Homeschools cannot be regulated/standardized.	15
Mainstream schools are of higher educational quality.	9
Con	
Parents should be permitted to choose the education of their children.	28
Homeschooling is often the best option for catering for the needs of exceptional/religious/ill/disabled students.	25
Homeschools can be personalized to the child's pace/needs.	21
Mainstream schools have a lot of violence/bullying.	21
The home is a good learning environment.	13
Parents will have more ability to pay-attention/educate their child.	7

Table 1: A sample key point-based summary, extracted from our ArgKP dataset.

ping arguments to these key points. This setting may be viewed as an intermediate step towards fully automatic argument summarization, but also as a valuable setting by itself: argument-to-key point mapping allows measuring the distribution of key points in a massive collection of arguments. It also allows interactive exploration of large argument collections, where key points serve as queries for retrieving matching arguments. In addition, it can be used for novelty detection - identifying unexpected arguments that do not match presupposed key points.

We develop the *ArgKP* dataset for the argument-to-keypoint mapping task, comprising about 24,000 *(argument, key point)* pairs labeled as matching/non matching.[1] To the best of our knowledge, this is the first dataset for this task. As discussed in the next section in more detail, our dataset is also much larger and far more comprehensive than datasets developed for related tasks such as mapping posts or comments in online debates to *reasons* or *arguments* (Hasan and Ng, 2014; Boltužić and Šnajder, 2014).

We report empirical results for an extensive set of supervised and unsupervised configurations, achieving promising results.

The main contributions of this work are:

1. We demonstrate, through extensive data annotation and analysis over a variety of topics, the feasibility and effectiveness of summarizing a large set of arguments collected from a large audience by mapping them to a small set of key points.

2. We develop the first large-scale dataset for the task of argument-to-key point mapping.

3. We perform empirical evaluation and analysis of a variety of classification methods for the above task.

2 Related Work

2.1 Argument Mining

The starting point for the current work is a collection of pro and con arguments for a given topic. As previously mentioned, these arguments may be collected from a large audience by conducting a survey, or mined automatically from text.

Some of the previous work on argument mining focused on specific domains such as legal documents (Moens et al., 2007; Wyner et al., 2010), student essays (Stab and Gurevych, 2017; Persing and Ng, 2016), and user comments on proposed regulations (Park and Cardie, 2014).

Mining arguments and argument components for a given topic (also known as *context*) has been a prominent line of research in argument mining. Levy et al. (2014) introduced the task of context-dependent claim detection in a collection of Wikipedia articles, and Rinott et al. (2015) did the same for context-dependent evidence detection. More recently, several works focused on topic-related argument mining from the Web or other massive corpora (Levy et al., 2017, 2018; Wachsmuth et al., 2017; Stab et al., 2018a,b; Ein-Dor et al., 2020).

Stance classification of extracted arguments can be performed as a separate step (Bar-Haim et al., 2017) or jointly with argument detection, as a three-way classification (pro argument/con argument/none), as done by Stab et al. (2018b).

---

[1] The dataset is available at `https://www.research.ibm.com/haifa/dept/vst/debating_data.shtml`

2.2 Argument Clustering and Summarization

Several works have focused on identifying pairs of similar arguments, or clustering similar arguments together. Ajjour et al. (2019) addressed the task of splitting a set of arguments into a set of non-overlapping *frames* such as *Economics*, *Environment* and *Politics*. Reimers et al. (2019) classified argument pairs as similar/dissimilar. Misra et al. (2016) aimed to detect argument pairs that are assumed to share the same *argument facet*, which is similar to our notion of *key points*. However, they did not attempt to explicitly identify or generate these facets, which remained implicit, but rather focused on detecting similarity between argument pairs. In contrast to these works, we directly map arguments to key points.

Egan et al. (2016) proposed to summarize argumentative discussions through the extraction of salient "points", where each point is a verb and its syntactic arguments. Applying their unsupervised method to online political debates showed significant improvement over a baseline extractive summarizer, according to human evaluation. While the current work also aims to summarize argumentative content via concise points, our goal is not to extract these points but to accurately map arguments to given points. Our main challenge is to identify the various ways in which the meaning of a point is conveyed in different arguments. The method employed by Egan et al. only matches arguments with the same *signature* - the same verb, subject and object dependency nodes, hence its ability to capture such variability is limited.

The line of work that seems most similar to ours is of Hasan and Ng (2014), Boltužić and Šnajder (2014) and Naderi (2016). Hasan and Ng classified posts and individual sentences from online debates into a closed set of *reasons*, composed manually for each topic. Boltužić and Šnajder mapped comments from one debating website (*ProCon.org*) to arguments taken from another debating website (*iDebate.org*). Naderi (2016) addressed a similar task: she used part of the Boltužić and Šnajder corpus as training data for an SVM classifier, which was then tested on sentences and paragraphs from same-sex marriage debates in the Canadian Parliament, annotated with the same set of arguments.

Our work differs from these works in several respects. First, we deal with crowd-contributed arguments, taken from the dataset of Gretz et al. (2020)

while these works dealt with posts or comments in debate forums, and parliamentary debates. Second, the dataset developed in this work is far more extensive, covering 28 topics and over 6,500 arguments[2], as compared to 2-4 topics in the datasets of Boltužić and Šnajder and Hasan and Ng, respectively. This allows us to perform a comprehensive analysis on the feasibility and effectiveness of argument-to-key point mapping over a variety of topics, which has not been possible with previous datasets. Lastly, while Hasan and Ng only perform within-topic classification, where the classifier is trained and tested on the same topic, we address the far more challenging task of cross-topic classification. Boltužić and Šnajder experimented with both within-topic and cross-topic classification, however they used a limited amount of data for training and testing: two topics, with less than 200 comments per topic.

Finally, we point out the similarity between the argument/key point relation and the text/hypothesis relation in *textual entailment*, also known as *natural language inference (NLI)* (Dagan et al., 2013). Indeed, Boltužić and Šnajder (2014) used textual entailment as part of their experiments, following the earlier work of Cabrio and Villata (2013), who used textual entailment to detect support/attack relations between arguments.

3 Data

3.1 Arguments and Key Points

As a source of arguments for this work we have used the publicly available IBM-Rank-30k dataset (Gretz et al., 2020). This dataset contains around 30K crowd-sourced arguments, annotated for polarity and point-wise quality. The arguments were collected with strict length limitations, accompanied by extensive quality control measures. Out of the 71 controversial topics in this dataset, we selected the subset of 28 topics for which a corresponding motion exists in the *Debatabase* repository of the *iDebate* website[3]. This requirement guaranteed that the selected topics were of high general interest.

We filtered arguments of low quality (below 0.5) and unclear polarity (below 0.6), to ensure sufficient argument quality in the downstream analysis. We randomly sampled 250 arguments per topic

---

[2] As detailed in the next section, a few hundreds of arguments out of the initial 7,000 were filtered in the process of constructing the dataset.

[3] https://idebate.org/debatabase

from the set of arguments that passed these filters (7,000 arguments in total for the 28 topics).

Debatabase lists several pro and con points per motion, where each point is typically 1-2 paragraphs long. The headline of each point is a concise sentence that summarizes the point. Initially, we intended to use these point headlines as our key points. However, we found them to be unsuitable for our purpose, due to a large variance in their level of specificity, and their low coverage of the crowd's arguments, as observed in our preliminary analysis.

To overcome this issue, we let a domain expert who is a professional debater write the key points from scratch. The expert debater received the list of topics and was asked to generate a maximum of 7 key points for each side of the topic, without being exposed to the list of arguments per topic. The maximal number of key points was set according to the typical number of pro and con points in Debatabase motions.

The process employed by the expert debater to produce the key points comprised several steps:

1. Given a debate topic, generate a list of possible key points in a constrained time frame of 10 minuets per side.

2. Unify related key points that can be expressed as a single key point.

3. Out of the created key points, select a maximum of 7 per side that are estimated to be the most immediate ones, hence the most likely to be chosen by crowd workers.

The process was completed within two working days. A total of 378 key points were generated, an average of 6.75 per side per topic.

3.2 Mapping Arguments to Key Points

3.2.1 Annotation Process

Using the Figure Eight crowd labeling platform[4], we created gold labels for associating the arguments selected as described in Section 3.1 with key points. For each argument, given in the context of its debatable topic, annotators were presented with the key points created for this topic in the relevant stance. They were guided to mark all of the key points this argument can be associated with, and if none are relevant, to select the 'None' option. Each argument was labeled by 8 annotators.

Quality Measures: to ensure the quality of the collected data, the following measures were taken -

1. Test questions. Annotators were asked to determine the stance of each argument towards the topic. Similarly to Toledo et al. (2019), this question functioned as a hidden text question[5]. All judgments of annotators failing in more than 10% of the stance questions were discarded.

2. Annotator-κ score. This score, measuring inter annotator agreement, as defined by Toledo et al. (2019), was calculated for each annotator, and all judgments of annotators with annotator-$\kappa < 0.3$ were ignored. This score averages all pair-wise Cohen's Kappa (Landis and Koch, 1997) for a given annotator, for any annotator sharing at least 50 judgments with at least 5 other annotators.

3. Selected group of trusted annotators. As in Gretz et al. (2020), the task was only available to a group of annotators which had performed well in previous tasks by our team.

As described above, the annotation of each key point with respect to a given argument was performed independently, and each annotator could select multiple key points to be associated with each given argument. For the purpose of calculating inter-annotator agreement, we considered *(argument, key point)* pairs, annotated with a binary label denoting whether the argument was matched to the key point. Fleiss' Kappa for this task was 0.44 (Fleiss, 1971), and Cohen's Kappa was 0.5 (averaging Annotator-κ scores). These scores correspond to "moderate agreement" and are comparable to agreement levels previously reported for other annotation tasks in computational argumentation (Boltužić and Šnajder, 2014; Ein-Dor et al., 2020). As for the stance selection question, 98% of the judgments were correct, indicating overall high annotation quality.

Data Cleansing: In addition to the above measures, the following annotations were removed from the data: (i) Annotations in which the answer to the stance selection question was wrong; (ii) Annotations in which key point choice was illegal - the 'None' option and one of the key points were

---

[4]http://figure-eight.com

both selected. However, the rate of these errors, for each of the annotators, was rather low ($< 10\%$ and $< 5\%$, respectively).

Arguments left with less than 7 valid judgments after applying the above quality measures and data cleansing were removed from the dataset. $6,568$ labeled arguments remain in the dataset.

3.2.2 Annotation Results

Next, we consolidate the individual annotations as follows. We say that an argument a is mapped to a key point k if at least 60% of the annotators mapped a to k. Recall that an argument can be mapped to more than one key point. Similarly, we say that a has *no key point* if at least 60% of the annotators mapped a to None (which is equivalent to not selecting any key point for the argument). Otherwise, we say that a is *ambiguous*, i.e., the annotations were indecisive. Table 2 shows examples for arguments and their matching key points in our dataset.

The distribution of the arguments in the dataset over the above categories is shown in Table 3. Remarkably, our key points, composed independently of the arguments, were able to cover 72.5% of them, with 5% of the arguments mapped to more than one key point.

We further investigated the differences between arguments in each category, by comparing their average quality score (taken from the IBM-Rank-30k dataset), number of tokens and number of sentences. The results are shown as additional columns in Table 3. Interestingly, arguments that have no key point tend to be shorter and have lower quality score, comparing to arguments mapped to a single key point; arguments mapped to more than one key point are the longest and have the highest quality.

Figure 1 examines the impact of the number of key points on argument coverage. For each topic and stance, we order the key points according to the number of their matched arguments, and add them incrementally. The results indicate that arguments are not trivially mapped to only one or two key points, but a combination of several key points is required to achieve high coverage. The marginal contribution decays for the sixth and seventh key points, suggesting that seven key points indeed suffice for this task.

22.8% of the arguments are *ambiguous*. Annotations for these arguments are split over several possible key points, none reaching the 60% threshold. For instance, the argument *"homeschooling*
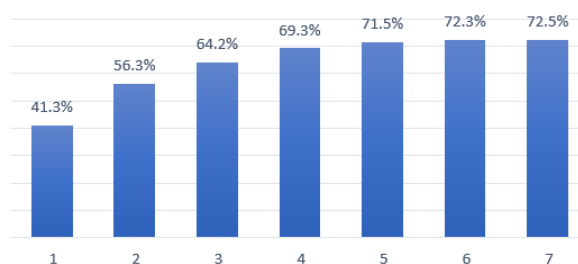

Figure 1: Argument coverage per number of key points.

enables parents with fringe views to push their agenda on their children without allowing exposure to alternative viewpoints.", had two key points with annotator votes higher than 40%, but below 60%:

1. *Homeschools cannot be regulated / standardized.*

2. *Parents are not qualified as teachers.*

Such cases suggest that many arguments are somewhat covered by the key points, but if the judgment is not clear-cut, the different intuitions of the annotators may result in no label receiving the required majority.

3.3 Final Dataset Generation

The *ArgKP* dataset includes *(argument, key point)* pairs with binary labels indicating whether the argument is matched to the key point. The dataset was created from the labeled data as follows. We define the *label score* of a pair as the fraction of annotations that classified the pair as matching . Pairs with label score ≥ 0.6 were labeled as positive (matching). Pairs with label score ≤ 0.15 were labeled as negative (non-matching). Pairs with label score in between these thresholds were removed.

We further cleansed our data by discarding key points having less than three matching arguments. This led to the removal of 135 out of the 378 key points and 14,679 out of 38,772 pairs obtained from the previous step.

The final dataset has 24,093 labeled *(argument, key point)* pairs, of which 4,998 pairs (20.7%) are positive. It has 6,515 arguments (232.67 per topic), and 243 key points (8.67 key points per topic). For each pair, the dataset also specifies the topic and the stance of the argument towards the topic.

We assessed the quality of the resulting dataset by having an expert annotator[6] mapping 100 ran-

---

[6]A professional debater who was not involved in the development of the dataset.

Topic	Argument	Associated Key Point(s)
We should end mandatory retirement.	Forcing members of a profession to retire at a certain age creates an experience drain.	A mandatory retirement age decreases institutional knowledge.
We should ban the use of child actors.	Child actors are fine to use as long as there is a responsible adult watching them.	Child performers should not be banned as long as there is supervision/regulation.
We should close Guantanamo Bay detention camp.	Guantanamo can provide security for accused terrorists who would be hurt in the general prison population.	The Guantanamo bay detention camp is better for prisoners than the alternatives.
Assisted suicide should be a criminal offence.	People have a basic right to bodily autonomy, deciding whether or not to die with minimal suffering and dignity is integral to that right.	People should have the freedom to choose to end their life. Assisted suicide gives dignity to the person that wants to commit it.
We should ban human cloning.	The world is already overpopulated, cloning humans will only contribute to this problem.	No key point

Table 2: Examples for key point association to arguments.

	% Arguments	Quality	# Tokens	# Sentences
No key point	4.7%	0.75	16.35	1.09
Ambiguous	22.8%	0.80	18.97	1.15
Single key point	67.5%	0.84	18.54	1.15
Multiple key points	5.0%	0.91	23.66	1.33

Table 3: Argument statistics by key point matches.

domly sampled arguments to key points, and comparing the annotations to the gold labels for all the corresponding pairs in the dataset. We obtained a remarkably high Cohen's Kappa of 0.82 ("almost perfect agreement"), validating the high quality of the dataset.

4 Experiments

4.1 Experimental Setup

We perform the task of matching arguments to key points in two steps. In the *Match Scoring* step (Section 4.1.1), we generate a score for each argument and key point. Then, in the *Match Classification* step (Section 4.1.2), we use these scores to classify the pairs as matching or non-matching.

We perform 4-fold cross-validation over the ArgKP dataset. Each fold comprises 7 test topics, 17 train topics and 4 development topics.

4.1.1 Match Scoring

We experimented with both unsupervised and supervised methods for computing a match score for a given *(argument, key point)* pair. We also explored transfer learning from the related task of natural language inference (NLI).

Unsupervised Methods

- **Tf-Idf.** In order to assess the role of lexical overlap in the matching task, we represent each argument and key point as tf-idf

weighted word vectors and use their cosine similarity as the match score.

- **Word Embedding.** We examined averaged word embeddings using GloVe (Pennington et al., 2014) and BERT (Devlin et al., 2019). GloVe is a context independent model that computes a single embedding for each word. BERT is a contextualized embedding model that takes the entire sentence into account. We also experimented with other embedding methods that under-performed BERT and thus their results are not reported here: Universal Sentence Encoder (Cer et al., 2018) and InferSent (Conneau et al., 2017). Again, we use cosine similarity to compute the match score.

Supervised Methods. We fine tuned the BERT-base-uncased and BERT-large-uncased models (Devlin et al., 2019) to predict matches between argument and key point pairs. We added a linear fully connected layer of size 1 followed by a sigmoid layer to the special [CLS] token in the BERT model, and trained it for three epochs with a learning rate of 2e-5 and a binary cross entropy loss.

NLI Transfer Learning. We also experimented with transfer learning from NLI to our task of argument-to-key point match classification. This was motivated by the similarity between these tasks (as discussed in Section 2.2), as well as the availability of large-scale NLI labeled datasets. We con-

sidered the Stanford (SNLI) and the Multi-Genre (MNLI) datasets (Bowman et al., 2015; Williams et al., 2018), each comprising hundreds of thousands of labeled premise-hypothesis pairs. Pairs labeled as ENTAILMENT were considered positive instances, while the rest of the pairs, labeled as NEUTRAL or CONTRADICTION were considered negative. We trained BERT-base and BERT-large models on each of these datasets, following the procedure described above.

4.1.2 Match Classification

In the match classification step we select the matching key points for each argument, based on their respective matching scores. The classification can be done locally, treating each pair individually, or globally, by examining all possible key points for each argument. We compared the following policies for selecting matching key points for a given argument.

Threshold. For each fold, we find the threshold on the match score that maximizes the F1 score for the positive (matching) class. Pairs whose score exceeds the learned threshold are considered matched.

Best Match (BM). Using a threshold is not optimal for our data, where most arguments have at most one matched key point. A natural solution is to select the best matching key point. For each argument, we consider all key points for the same topic and stance as candidates and predict only the candidate with the highest match score as matched to the argument and the rest as unmatched. Note that this is the only fully unsupervised selection policy, as it does not require labeled data for learning a threshold.

BM+Threshold. The *BM* policy always assigns exactly one key point for each argument, while 27.5% of the arguments in our data are not matched to any key point. To address this, we combine the two former policies. The top matching key point is considered a match only if its match score exceeds the learned threshold.

Dual Threshold. In order to account for arguments with more than one matching key point, two thresholds are learned. If two key points exceed the lower threshold and at least one of them exceeds the upper threshold, both will be matched. Otherwise, it works the same as the *BM+Threshold*

policy using only the lower threshold. This allows for zero to two matches per argument.

Thresholds are learned from the development set for supervised match scoring methods, and from both train and development set for unsupervised match scoring methods.

4.2 Results

4.2.1 Match Scoring Methods

Table 4 compares the various match scoring methods, all using the *Threshold* key point selection policy. Results are obtained by micro-averaging over the argument-key point pairs in each fold, and averaging over the different folds. We consider Precision, Recall and F1 of the positive class, as well as the overall accuracy. We also list for reference the majority class baseline that always predicts "no match", and the random baseline, which randomly predicts the positive class according to its probability in the training data.

The unsupervised models fail to capture the relation between the argument and the key points. Tf-Idf and Glove perform the worst, showing that simple lexical similarity is insufficient for this task. BERT embedding does better but still reaches a relatively low F1 score of 0.4.

In contrast to the unsupervised models, supervised models are shown to perform well. BERT with fine tuning leads to a substantial improvement, reaching F1 score of 0.657 with the BERT-base model, and 0.684 with the BERT-large model.

BERT Models trained on NLI data are considerably better than the unsupervised methods, with the best model reaching F1 of 0.526, yet their performance is still far below the supervised models trained on our ArgKP dataset. This may reflect both the similarities and the differences between NLI and the current task. We have also experimented with combining these two types of data in cascade: BERT was first trained on a large NLI dataset (SNLI, MNLI or their union), and was then fine-tuned on the smaller ArgKP data. However, it did not improve the supervised results.

Error Analysis. By analyzing the top errors of the supervised classifier (BERT-large), we found several systematic patterns of errors. In most cases, non-matching arguments and key points received a high match score in one of the following cases:

- They share some key phrases. For example: *"It is unfair to only subsidize vocational education. Achieving a more advanced education*

		Acc	P	R	F1
	Majority Class	0.793		0.000	
	Random Predictions	0.679	0.206	0.200	0.203
Unsupervised Methods	Tf-Idf	0.512	0.246	0.644	0.352
	Glove Embeddings	0.346	0.212	0.787	0.330
	BERT Embeddings	0.660	0.319	0.550	0.403
Supervised Methods	BERT-base (ArgKP)	0.844	0.609	0.718	0.657
	BERT-large (ArgKP)	**0.868**	**0.685**	**0.688**	**0.684**
NLI Transfer Learning	BERT-base (SNLI)	0.777	0.472	0.514	0.485
	BERT-base (MNLI)	0.772	0.470	0.558	0.505
	BERT-large (SNLI)	0.765	0.456	0.533	0.487
	BERT-large (MNLI)	0.792	0.518	0.542	0.526

Table 4: Comparison of match scoring methods, using the *Threshold* selection policy. P, R and F1 refer to the positive class. Acc is the accuracy.

	All Arguments				Single Key Point			Multiple Key Points			No Key Points
	Acc	P	R	F1	P	R	F1	P	R	F1	Acc
Threshold	.868	.685	.688	.684	.720	.686	.701	.904	**.690**	**.782**	**.933**
Best Match	.876	.696	.711	.703	.836	.747	**.789**	.936	.448	.606	.839
BM+Threshold	**.890**	**.772**	.665	.713	**.856**	.699	.769	.941	.421	.580	.915
Dual Threshold	.887	.721	**.740**	**.730**	.784	**.752**	.767	**.945**	.656	.773	.908

Table 5: Comparing key point selection policies, using BERT-large trained on the ArgKP dataset for match scoring.

is very expensive and it would also need to be subsidized." and *"Subsidizing vocational education is expensive"*.

- They share a large portion of the sentence, but not the main point, for example: *"Women should be able to fight if they are strong enough"* and *"Women should be able to serve in combat if they choose to"*.

- They are at least partially related, but labeled as non-matching due to a better fitting key point for the same argument. For example: *"We should subsidize space exploration because it increases the knowledge of the universe we are in"* and *"Space exploration improves science/technology"* can be considered matched, but were labeled as unmatched due to the key point *"Space exploration unravels information about the universe"*. Using the *Best Match* policy helps in these cases.

For arguments and key points that were labeled as matched but received a low match score, the relation was in many cases implied or required some further knowledge, for examples: *"Journalism is an essential part of democracy and freedom of expression and should not be subsidized by the state."* and *"government intervention has the risk of inserting bias/harming objectivity"*.

4.2.2 Key Point Selection Policies

Table 5 compares different key point selection policies, all using the best performing match scoring method: BERT-large fine-tuned on ArgKP. We report the results over the whole dataset ("all arguments"), as well as the subsets of arguments having none, single or multiple matching key points according to the labeled data. In case of no matches we present the accuracy, as recall and F1 scores are undefined. When considering all the arguments, the *Dual Threshold* policy achieves the best F1 score of 0.73. The *Threshold* method performs well for arguments with no matches or multiple matches. When there is exactly one match (the common case in our data), it has lower precision. The *Best Match* policy performs well when there is a single match, but is not able to cope with arguments that have no matches or have multiple matches. The *BM+Threshold* method combines the two and is useful when there are no matching key points or a single matching key point, but still have lower recall when there are multiple matching key points. The *Dual Threshold* method improves the recall and therefore the F1 score for multiple matches while maintaining good performance for arguments with single or no matches.

Figure 2 shows Precision-Recall trade-off for the various policies, using the different possible thresholds, computed for one of the folds. For each policy, we specify the best F1 score, as well
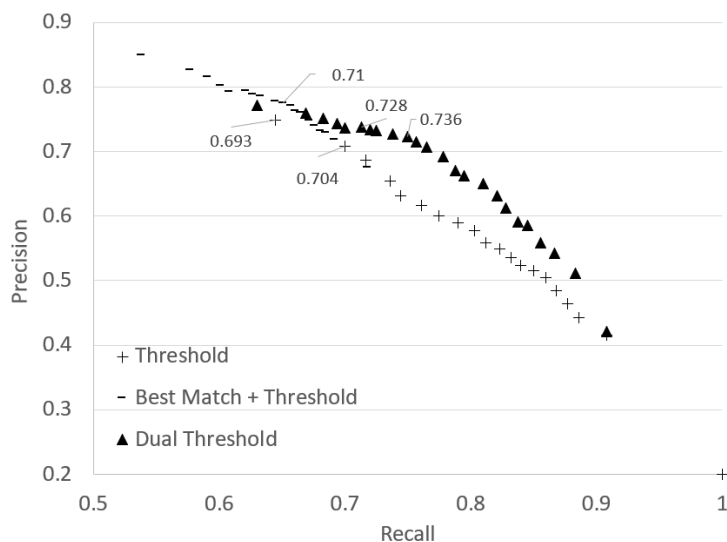
Figure 2: Precision/Recall trade-off for different key point selection policies. For each method, the highest F1 score, as well as the F1 score for the chosen threshold are specified. For the *Best Match + Threshold* policy, these two scores coincide.

as the F1 score obtained for the selected threshold, which was optimized over the development set. The *Threshold* policy allows to control recall, up to one (where the threshold is zero), at the price of low precision. The *BM+Threshold* policy generates the highest precision, but low recall, since at most one candidate is selected. Note that when the threshold is zero, the *BM+Threshold* policy is equivalent to the *BM* policy. The *Dual Threshold* policy offers the best trade-off, for mid-range precision and recall.

5 Conclusion

This work addressed the practical problem of summarizing a large collection of arguments on a given topic. We proposed to represent such summaries as a set of key points scored according to their relative salience. Such summary aims to provide both textual and quantitative views of the argument data in a concise form. We demonstrated the feasibility and effectiveness of the proposed approach through extensive data annotation and analysis. We showed that a domain expert can quickly come up with a short list of pro and con key points per topic, that would capture the gist of crowd-contributed arguments, even without being exposed to the arguments themselves. We studied the problem of automatically matching arguments to key points, and developed the first large-scale dataset for this task, which we make publicly available.

Our experimental results demonstrate that the

problem is far from trivial, and cannot be effectively solved using unsupervised methods based on word or sentence-level embedding. However, by using state of the art supervised learning methods for match scoring, together with an appropriate key point selection policy for match classification, we were able to achieve promising results on this task.

The natural next step for this work is the challenging task of automatic key point generation. In addition, we plan to apply the methods presented in this work also to automatically-mined arguments. Finally, detecting the more implicit relations between the argument and the key point, as seen in our error analysis, is another intriguing direction for future work.

Acknowledgments

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.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance

classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261. Association for Computational Linguistics.

Filip Boltužić and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Elena Cabrio and Serena Villata. 2013. A natural language bipolar argumentation approach to support users in online debate interactions. *Argument & Computation*, 4(3):209–230.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

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.

Charlie Egan, Advaith Siddharthan, and Adam Wyner. 2016. Summarising the points made in online political debates. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 134–143, Berlin, Germany. Association for Computational Linguistics.

Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. 2020. Corpus wide argument mining – a working solution. In *AAAI*.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. In *AAAI*.

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.

J. R. Landis and G. G. Koch. 1997. The measurements of observer agreement for categorical data. *Biometrics*, 33:159–174.

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

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Ran Levy, Ben Bogin, Shai Gretz, Ranit Aharonov, and Noam Slonim. 2018. Towards an argumentative content search engine using weak supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2066–2081. Association for Computational Linguistics.

Ran Levy, Shai Gretz, Benjamin Sznajder, Shay Hummel, Ranit Aharonov, and Noam Slonim. 2017. Unsupervised corpus–wide claim detection. In *Proceedings of the 4th Workshop on Argument Mining*, pages 79–84. Association for Computational Linguistics.

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.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ICAIL '07, pages 225–230, New York, NY, USA. ACM.

Nona Naderi. 2016. Argumentation mining in parliamentary discourse. In *Proceedings of the 11th International Conference of the Ontario Society for the Study of Argumentation*, pages 1–9.

Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. 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.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal. Association for Computational Linguistics.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018a. Argumentext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018b. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674. Association for Computational Linguistics.

Assaf Toledo, Shai Gretz, Edo Cohen-Karlik, Roni Friedman, Elad Venezian, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2019. Automatic argument quality assessment - new datasets and methods. 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 5624–5634, Hong Kong, China. Association for Computational Linguistics.

Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. 2010. Approaches to text mining arguments from legal cases. In *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, page 60–79, Berlin, Heidelberg. Springer-Verlag.