Extractive Conversation Summarization Driven by Textual Entailment Prediction

Extractive Conversation Summarization Driven by Textual Entailment Prediction

Giuseppe Gallipoli
Dip. di Automatica e Informatica
Politecnico di Torino
Turin, Italy
giuseppe.gallipoli@polito.it

Luca Cagliero
Dip. di Automatica e Informatica
Politecnico di Torino
Turin, Italy
luca.cagliero@polito.it

Paolo Garza
Dip. di Automatica e Informatica
Politecnico di Torino
Turin, Italy
paolo.garza@polito.it

***Abstract*—Summarizing conversations like meetings, email threads or discussion forums poses relevant challenges on how to model the dialogue structure. Existing approaches mainly focus on premise-claim entailment relationships while neglecting contrasting or uncertain assertions. Furthermore, existing techniques are abstractive, thus requiring a training set consisting of humanly generated summaries. With the twofold aim of enriching the dialogue representation and addressing conversation summarization in the absence of training data, we present an extractive conversation summarization pipeline. We explore the use of contradictions and neutral premise-claim relations, both in the same document or in different documents. The results achieved on four datasets covering different domains show that applying unsupervised methods on top of a refined premise-claim selection achieves competitive performance in most domains.**

***Index Terms*—Conversation Summarization, Extractive Summarization, Textual Entailment, Conversational AI**

I. Introduction

The diffusion of social networks, video communication platforms, and online dialogue systems has fostered the generation of abundant amounts of conversational data. They include business meeting conversations, email threads, community question answering, and discussion forums on newsworthy topics. Gaining an insight into these textual contents can be extremely time-consuming, thus deteriorating the user experience. To bridge this gap, summarization techniques have been used to extract the most salient dialogues' content [1]–[3].

Compared to traditional approaches designed for news or timeline summarization (e.g., [4], [5]), conversation summarization techniques need to face the following issues:

- **Discourse structure**: Traditional summarizers are mainly designed to handle free text. The dialogue structure is considerably different as the underlying message and context should be inferred from the speakers' interactions.
- **Multiple speakers and turn-taking**: Dialogues involve multiple participants that take turns to speak or respond to each other. Conversely, traditional documents usually have a single writer (or speaker).
- **Language style**: Unlike most written documents, the style of dialogue reflects the language style used in human communication, either informal or domain-specific, according to the context.
- **Presence of noise and fragmented data**: Conversational data can be noisy, contain non-informative utterances or fragmented pieces of information. The impact of redundancy and fragmentation is likely to be more severe than in traditional scenarios such as news summarization.

To consider the discourse structure, recent works incorporate dialogue-discourse relations and model dialogue interactions using graphs [1]–[3]. To handle multiple speakers, other studies focus on modeling dialogue participants [6], [7]. A recent on-topic survey can be found in [8].

The present work aims at defining an effective representation of the dialogue text. State-of-the-art approaches (e.g., [3]) first model premise-claim relationships among dialogue sentences in a graph and then extract the entailment relationships separately for each document. Therefore, they neglect contradictions between sentences belonging to different documents. To the best of our knowledge, all the existing approaches to conversation summarization are abstractive [8]. Hence, they need humanly generated conversation summaries. However, annotating a conversation dataset is labor-intensive and pre-trained models can be unavailable for specific dialogue and language types.

This paper addresses the main limitations of existing conversation summarization methods regarding dialogue representation and the need for annotated data. We perform multi-document textual entailment classification to enhance the sentence-level representation of the dialogue content. Specifically, we propose to also leverage the contradictory and neutral relationships. For example, the following pair of sentences *I like the entertainment services* and *I will never use the entertainment services again* expresses a contrasting opinion that is likely to appear in real-life conversations.

To enable conversation summarization in the absence of a sufficiently large number of reference (humanly generated) summaries, we adopt an unsupervised extractive pipeline. In particular, instead of training a sequence-to-sequence model on the annotated conversation summaries, we first shortlist the most relevant sentences leveraging entailment classification and then apply established summarization methods in a multi-document setting.

The main contributions of our work are outlined below:

- **Multi-document entailment classification**: Unlike prior works, we analyze textual entailments in a multi-document dialogue representation and consider the contradiction and neutral relationships to handle contrasting or uncertain opinions.
- **Extractive, unsupervised summary generation**: We use extractive summarization techniques that do not necessarily require training examples of conversation summaries.
- **Ablation study and algorithms' comparison**: We test multiple summarization techniques and dialogue text representations on benchmark conversation datasets [3]. The best-performing approach achieves great improvements compared to the baselines (i.e., +3.8 ROUGE-1 F1-score).

II. Preliminaries

We formulate the *conversation summarization task* as a multi-document summarization problem. Given a conversation $\mathcal{D}$=$\{D_1, D_2, ..., D_n\}$ composed of n documents D_i, the purpose is to summarize $\mathcal{D}$. Each document corresponds to a self-contained snippet of the dialogue text and refers to a user.

For example, let us consider a discussion thread in an online forum. Each document consists of a separate user reply. Without any loss of generality, we can assume that each reply reflects only the author's opinion and is not necessarily in agreement with those of the other users.

An *extractive summary S* consists of a shortlist of document sentences $s_j \in \bigcup_1^n D_i$. Conversely, an *abstractive summary* contains sentences that do not necessarily appear in $\mathcal{D}$. In this work, we specifically address extractive summarization.

To define the key *argumentative units* in each conversation, we segment the input documents' text into sentences and identify the following sentence types [9]: *claim*, i.e., an assertion that something is true, and *premise*, i.e., the proposition based on which we can make a claim assertion.

For our purposes, non-argumentative sentences are early pruned and the premise-claim pairs extracted from the documents are deemed as the textual units that are most discriminating for summary generation.

III. Method

Figure 1 depicts the proposed methodology to address extractive conversation summarization. Given the input conversation $\mathcal{D}$, it first extracts the main arguments, which consist of pairs of premises and claims, and discards non-argumentative sentences (see Step (1)).

Step (2) enriches the dialogue representation by automatically classifying the relationship holding between the premise and claim sentences. The types include not only entailment relationships, i.e., the claim supports the premise, but also contradictions and neutral relationships, i.e., the content of the claim and the premise is contrasting or not necessarily related.

Step (3) filters the input document content based on the outcomes of Steps (1) and (2). The goal is to keep only the sentence-level snippets of dialogue text that are potentially worth including in the summary by covering the most relevant argumentative units and sentence relationships.

Finally, Step (4) applies extractive summarization on top of the filtered sentences. The extractive procedure shortlists the sentences returned by Step (3) to compose the output summary.

More details on each step are given in the following.

A. Argument Extraction

After segmenting the input text into sentences, the document units are classified as *premise* (*P*), *claim* (*C*) or *non-argumentative* (*NA*). To this end, similar to [3], we utilize a BERT-based argument extraction module [10].

The sentences classified as argumentative units, i.e., either *P* or *C*, are kept. Conversely, the non-argumentative units are early discarded as long as a minimal portion of the original document content is preserved. The key idea is to discard the redundant sentences without excessively reducing the number of assertions in the conversation document. We handle the above exceptions by using the heuristics adopted by [3].

B. Relationship Type Classification

To model the dialogue structure, we analyze the relations between dialogue units in the conversation data at the level of premise-claim pairs.

We generate all the possible premise-claim pairs $\langle P, C \rangle$ and then perform textual entailment classification to predict the type of relationship holding between P and C. To accomplish this task, we adopt a RoBERTa [11] classifier fine-tuned on the MNLI entailment dataset. Sentence pairs can be labeled as *entailment*, *contradiction* or *neutral*.

Entailment relationships hold when the two sentences are in agreement with each other, e.g., *Smoke comes out of the windshield* and *The car engine is damaged*. Contradiction relationships hold when the two statements are contrasting, e.g., *Smoke comes out of the windshield* and *The car engine is turned off*. All the remaining cases belong to the neutral relationship type.

Unlike state-of-the-art conversation summarization approaches (e.g., [3]), which ignore contradiction and neutral relationship types, our approach adopts a more conservative strategy. Specifically, all three relationship types (i.e., entailment, contradiction and neutral) are maintained.

Neutral relationships represent pairs of weakly correlated sentences. They may represent either pairs of assertions that are anyhow worth considering separately (hereinafter denoted by *keep* neutral setting), ambiguous premise-claim relationships that should be reassigned to either the entailment or contradiction categories (hereinafter denoted by *binary* neutral setting), or combinations of sentences that are misleading and thus should be disregarded (hereinafter denoted by *remove* neutral setting). An empirical analysis of the different settings for the neutral relationships is given in Section IV.

Notice that within the same document premises and claims are likely to be concordant or neutral. Conversely, contradictions likely emerge in different documents, revealing contrasting opinions. With the goal of including complementary views and opinions in the conversation summary, we consider the following scenarios: *intra-document*, which considers only

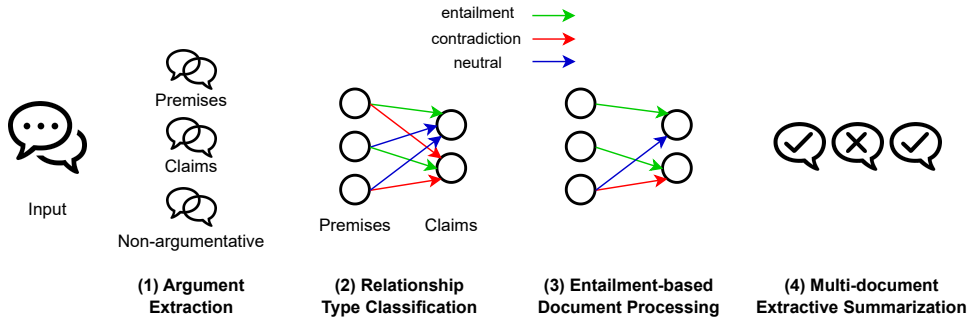

Fig. 1. The proposed pipeline for conversation summarization.

the premise-claim relationships within the same document of the conversation, and *inter-document*, which considers also the premise-claim relationships among different documents.

C. Entailment-based Document Processing

Since many premise-claim pairs are potentially redundant, the goal is to filter out irrelevant pairs based on the relationship type classification. Premise-claim relationships are potentially many-to-many (i.e., each premise can potentially be associated with multiple claims and vice versa), therefore we envision the following content filtering strategies:

- *one_premise*: Among the claims associated with the same premise, we keep only the one that maximizes the textual entailment classification score. Notice that a claim could still appear in multiple selected pairs (associated with different premises).
- *one_claim*: Among the premises associated with the same claim, we keep only the one maximizing the textual entailment classification score. Notice that a premise could still appear in multiple selected pairs (associated with different claims).
- *one_premise_claim*: Among the pairs sharing either the same premise or claim, we keep only the one maximizing the textual entailment classification score.
- *no_filter*: We keep the full set of premise-claim pairs.

The filtered sentences are provided as input to the next stage. Since most of the summarization techniques considered operate at the sentence level, it is possible to regard the juxtaposition of the premise and claim from the same pair as a single sentence or treat them as two independent sentences.

D. Multi-document Extractive Summarization

This step takes as input the sentences selected in the previous step and produces a unified summary consisting of a shortlist of the top-k most relevant sentences. The purpose is to condense the original content into a non-redundant selection of the most relevant argumentative units.

The summary may include either a selection of the $\frac{k}{2}$ premise-claim sentence pairs or an arbitrary combination of k distinct sentences (disregarding the juxtaposition of premises and claims in the dialogue representation).

Among the existing extractive summarization methods, we considered five established methods. However, the proposed pipeline can straightforwardly support additional methods. Three out of five selected summarizers are *unsupervised*, i.e., they process the input documents without requiring any reference summaries. Their use is particularly appealing in the absence of training data for the dialogue type and language under analysis. For the sake of completeness, we also explore the use of *supervised* extractive methods, which shortlist the input sentences based on a predictive model trained on an annotated set of documents. The use of abstractive summarization methods is out of the scope of this work.

We consider the following unsupervised approaches:

TextRank [12]: It is an established graph-based summarization method. The input text is modeled as a graph whose vertices are sentences and edges are weighted according to a similarity measure. Vertices are ranked by the PageRank [13] algorithm and the top-k top ranked sentences are returned.

LexRank [14]: It is another established graph-based method. It constructs a similarity matrix containing the similarity scores between each pair of sentences. Then, it prunes the sentence links based on a threshold and assigns an importance score to each sentence based on the concept of eigenvector centrality. The top-k most important sentences are returned.

Clustering: To identify groups of similar sentences and select the best representatives, we also consider the following established clustering algorithms: K-Means and DBSCAN. Specifically, we cluster the premise-claim pairs based on their embedding representations of the `[CLS]` token encoded by the textual entailment classifier. Then, for each cluster we extract a subset of representative pairs by minimizing the cumulative pairwise distance.

We also consider the following supervised approaches:

BERTSum [15]: It is a BERT-based architecture for text summarization. It extends BERT with additional layers which are fine-tuned for a sentence classification task. For each sentence, the goal is to decide whether the sentence is worth being included in the summary or not. The top-k sentences are shortlisted according to the classification score.

MatchSum [4]: On top of BERTSum output, we consider an unsupervised variant of a state-of-the-art approach based on text matching. It first extracts the x most salient input sentences using BERTSum ($x > k$). Then, it generates all the $\binom{x}{k}$ combinations of k sentences. Each combination represents a candidate summary of the conversation. MatchSum compares

TABLE I
CONVOSUMM BENCHMARK – DATASET STATISTICS: AVERAGE NUMBER OF TOKENS AND DOCUMENTS PER SAMPLE.

dataset	input length	sentence length	summary length	# docs
NYT	1646	22	87	18.1
Reddit	588	18	73	13.7
Stack	822	30	83	6.3
Email	1237	45	87	4.9

the embedding of each candidate summary with the input text and selects the most similar candidate according to the cosine similarity. The key idea is to choose the candidate summary that maximizes the coverage of the whole documents' content.

For TextRank, LexRank, BERTSum, and MatchSum, we also test a baseline version without textual entailment classification. The idea behind it is to provide the summarizer with the raw dialogue text, rather than the premise-claim pairs, and compare the performance with that of the corresponding entailment-enriched version (for the same value of top-k). For the clustering techniques, the raw text is encoded using its embedding representation.

IV. EXPERIMENTAL RESULTS

We run the experiments on a single machine equipped with an NVIDIA® V100 GPU with 32 GB of VRAM.

A. Dataset

We utilize the recently proposed ConvoSumm benchmark [3]. It consists of four datasets ranging over the following domains: news article comments (i.e., NYT), discussion forum debates (i.e., Reddit), community question answering (i.e., Stack), and email threads (i.e., Email). For each domain, there are 200/50/250 train/validation/test samples, respectively, each paired with a manually annotated abstractive summary. We consider the validation set for hyperparameter tuning and the test set for computing the performance metrics.

Table I reports the main dataset statistics. They show a quite high variability in document cardinality, document length, and sentence characteristics. This confirms the importance of adopting multi-document dialogue representations.

Samples are enriched with domain-specific metadata (e.g., article headline, answer score), which are removed when reading data. Some domains also include contextual information (e.g., article snippet, forum post), which is disregarded for summary generation.

B. Evaluation metrics

We evaluate our summarization methods by means of the widely used ROUGE metrics [16]. They count the syntactic unit overlap between the output summaries and the reference ones. We consider the ROUGE-1/2/L F1-scores. As an additional semantic metric, we use the BERTScore F1-score [17].

C. Algorithm implementations

For textual entailment classification, we rely on the RoBERTa-large model available in the Hugging Face Transformers library. For TextRank and LexRank, we use the implementations available in the `summa`[1] and `lexrank`[2] packages, respectively. For BERTSum, we use the official checkpoint fine-tuned on the CNN/DailyMail dataset, whereas for MatchSum the candidate summaries and input texts are encoded using RoBERTa-base. For the clustering approaches, we use the `scikit-learn` library and SentenceBERT [18] as text encoder based on the `all-MiniLM-L6-v2` model.

D. Hyperparameter tuning

Separately for each dataset domain, we choose the best configuration for each algorithm according to the ROUGE-1 F1-score achieved on the validation set.

We focus on the following summarizer-agnostic hyperparameters: (1) raw documents (no entailment classification) vs. entailment-only (disregarding contradictions and neutral pairs) vs. entailment-enriched (considering entailment, contradiction and neutral relationships); (2) intra- vs. inter-document (see Section III-B); (3) one_premise vs. one_claim vs. one_premise_claim vs. no_filter (see Section III-C); (4) keep vs. binary vs. remove (see Section III-B); (5) juxtaposition of premise-claim sentences vs. separate sentences (see Section III-C).

We also performed a grid search over the following summarizer-specific hyperparameters:

- *Clustering*: We varied the number of clusters and the distance metric used by the clustering algorithm.
- *MatchSum*: We set x to 10 and experiment with three truncation methods to encode the input text. Let *max_len* be the maximum length allowed by the encoder model, we consider the first/middle/last *max_len* tokens.

In general, for each algorithm and dataset domain, we conduct experiments with multiple top-k ranging between 3 and 8.

E. Results

Tables II(a)-(d) report the results achieved on the NYT, Reddit, Stack, and Email datasets, respectively. In the NYT domain, the TextRank algorithm with k=5 performs best (see Table II(a)). Notably, it yields a significant performance improvement (+1 ROUGE-1 F1-score) against its baseline version using the raw text. This confirms the effectiveness of the entailment-based dialogue representation. According to BERTScore, the best performing method is LexRank, which is again an unsupervised graph-based model. Focusing on the results achieved on the Reddit social posts, MatchSum with k=7 performs best and outperforms its baseline version (see Table II(b)). However, the performance gap with TextRank is quite limited. MatchSum and LexRank perform best in terms of BERTScore, confirming the applicability of unsupervised models. On the Stack dataset, BERTSum and LexRank perform equally best. Again, most of the tested approaches outperform the corresponding baseline versions (see Table II(c)). Considering the results on the Email domain in Table II(d), BERTSum with k=6 consistently outperforms

[1]https://pypi.org/project/summa/ – Last access: July 2023
[2]https://pypi.org/project/lexrank/ – Last access: July 2023

both the baseline and the other algorithms for all the evaluation metrics (e.g., +1.6 ROUGE-1 F1-score improvement). The ROUGE scores achieved on Email are, on average, superior to those obtained in the other domains. This is likely due to the conciseness of the email dialogues, i.e., the more limited number of input documents to be processed (see Table I).

The overall results demonstrate the effectiveness of our approach with respect to the baseline methods. In most cases, the entailment-based approach achieves improvements of more than +1 ROUGE-1 F1-score and up to +3.8 points on the Email domain. The only exception is clustering-based summarization, where the baseline performs better than the entailment-based approach (even though clustering is not the top performer in any of the analyzed domains). Notice that since clustering algorithms are applied to the encoded sentences and not to their raw version, the clustering baseline is actually not directly comparable with the other ones.

F. Hyperparameter analysis

Based on the results reported in Tables II(a)-(d), we can draw the following takeaways:

a) Representation: The combined use of all relationship types significantly improves the performance compared to the raw and entailment-only versions, confirming the relevance of contradictory and neutral opinions.

b) Intra- vs. inter-document: Focusing on intra-document premise-claim pairs only is beneficial as the number of spurious combinations is lower.

c) Pair method: The most effective method to filter sentence pairs is *one_claim*, which entails modeling premise-claim pairs as many-to-one relationships. This is likely due to the fact that many premises yield the same claim, thus *one_claim* reduces the redundancy in conversation data. Claims are also more likely to contain the key information in the dialogues.

d) Neutral: Keeping neutral pairs or reassigning them to the entailment or contradiction categories is beneficial since they may contain valuable information that would otherwise be neglected if they were discarded.

e) Juxtaposition: Keeping premises and claims separated is always beneficial, likely because premise content is highly redundant and, in several cases, can be omitted.

V. Conclusions and future extensions

We presented a novel extractive approach to conversation summarization. We show that enriching textual entailment classification with contradiction and neutral relationships can improve the quality of the resulting summaries on most dialogue data types. We also compare the performance of unsupervised and supervised summarization methods, showing that graph-based methods can achieve promising performance even in the absence of training data.

As future work, we plan to further enrich the representation of the dialogue structure and leverage cross-type dialogue relations (e.g., Reddit conversations and email threads). We also plan to address abstractive conversation summarization.

Acknowledgment

This study was carried out within the MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from Next-GenerationEU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1551.11-10-2022, PE00000004). This study was also partially carried out within the FAIR (Future Artificial Intelligence Research) and received funding from Next-GenerationEU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1555.11-10-2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] J. Chen and D. Yang, "Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization," in *Proceedings of EMNLP'20*. ACL, 2020, pp. 4106–4118.

[2] X. Feng, X. Feng, B. Qin, and X. Geng, "Dialogue discourse-aware graph model and data augmentation for meeting summarization," in *IJCAI'20*, 2020.

[3] A. Fabbri, F. Rahman, I. Rizvi, B. Wang, H. Li, Y. Mehdad, and D. Radev, "ConvoSumm: Conversation summarization benchmark and improved abstractive summarization with argument mining," in *Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNLP (Volume 1: Long Papers)*. ACL, 2021, pp. 6866–6880.

[4] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," in *Proceedings of the 58th Annual Meeting of the ACL*. ACL, 2020, pp. 6197–6208.

[5] M. La Quatra, L. Cagliero, E. Baralis, A. Messina, and M. Montagnuolo, "Summarize dates first: A paradigm shift in timeline summarization," in *Proceedings of SIGIR '21*. ACM, 2021, p. 418–427.

[6] Y. Lei, Y. Yan, Z. Zeng, K. He, X. Zhang, and W. Xu, "Hierarchical speaker-aware sequence-to-sequence model for dialogue summarization," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7823–7827, 2021.

[7] Z. Liu and N. Chen, "Controllable neural dialogue summarization with personal named entity planning," in *Proceedings of EMNLP'21*. ACL, 2021, pp. 92–106.

[8] X. Feng, X. Feng, and B. Qin, "A survey on dialogue summarization: Recent advances and new frontiers," in *Proceedings of IJCAI-22*. IJCAI Organization, 2022, pp. 5453–5460.

[9] M. Lenz, P. Sahitaj, S. Kallenberg, C. Coors, L. Dumani, R. Schenkel, and R. Bergmann, "Towards an Argument Mining Pipeline Transforming Texts to Argument Graphs," in *Proceedings of the 8th International Conference on Computational Models of Argument*, ser. Frontiers in Artificial Intelligence and Applications, vol. 326, 2020, pp. 263–270.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT'19*. ACL, 2019, pp. 4171–4186.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[12] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of EMNLP'04*. ACL, 2004, pp. 404–411.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking : Bringing order to the web," in *The Web Conference*, 1999.

[14] G. Erkan and D. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research - JAIR*, vol. 22, 2011.

[15] Y. Liu, "Fine-tune BERT for extractive summarization," *CoRR*, vol. abs/1903.10318, 2019.

[16] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. ACL, 2004, pp. 74–81.

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with BERT," *CoRR*, vol. abs/1904.09675, 2019.

[18] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of EMNLP-IJCNLP'19*. ACL, 2019, pp. 3982–3992.

TABLE II

RESULTS ON THE NYT, REDDIT, STACK, AND EMAIL DATASETS – ROUGE-1/2/L (R1/2/L) AND BERTSCORE (BS) F1-SCORES. **BOLD** AND UNDERLINE RESPECTIVELY DENOTE THE BEST SCORE FOR EACH METRIC SEPARATELY FOR EACH ALGORITHM AND ACROSS DIFFERENT ALGORITHMS.

algorithm	representation	intra-document	pair_method	neutral	juxtaposition	k	R1	R2	RL	BS
TextRank	raw	–	–	–	–	5	24.4	4.0	13.5	83.9
	entailment-enriched	True	one_claim	keep	False	5	**25.4**	**4.5**	**14.2**	**84.2**
	entailment-only	True	no_filter	–	False	5	18.3	2.7	11.0	78.0
LexRank	raw	–	–	–	–	6	23.7	3.9	13.9	84.2
	entailment-enriched	True	one_claim	keep	False	6	**24.6**	**4.3**	**14.0**	**84.3**
	entailment-only	True	no_filter	–	False	6	18.6	2.7	11.2	79.5
BERTSum	raw	–	–	–	–	6	23.8	3.2	13.2	83.7
	entailment-enriched	True	one_claim	keep	False	6	**24.7**	**3.8**	**13.5**	**84.0**
	entailment-only	True	no_filter	–	False	6	19.4	2.9	11.5	79.8
K-Means	raw	–	–	–	–	8	**24.4**	**3.9**	**13.0**	83.8
	entailment-enriched	False	one_p_c	binary	–	8	23.1	3.1	12.8	**83.9**
	entailment-only	True	one_premise	–	–	8	16.5	2.2	10.4	78.4
DBSCAN	raw	–	–	–	–	8	**24.2**	**4.1**	**13.2**	**83.7**
	entailment-enriched	True	one_p_c	binary	–	8	23.0	3.1	12.5	83.6
	entailment-only	True	no_filter	–	–	8	13.1	1.8	8.2	58.7
MatchSum	raw	–	–	–	–	5	24.1	3.2	13.2	83.7
	entailment-enriched	True	one_claim	keep	False	5	**25.2**	**3.6**	**13.8**	**84.0**
	entailment-only	True	one_premise	–	False	5	18.8	2.7	11.4	78.7

(a) NYT domain

TextRank	raw	–	–	–	–	6	22.8	4.1	13.1	83.6
	entailment-enriched	True	one_claim	keep	False	6	**23.2**	**4.4**	**13.4**	**83.8**
	entailment-only	True	no_filter	–	False	6	19.5	3.3	11.7	82.1
LexRank	raw	–	–	–	–	7	22.9	4.3	13.2	83.9
	entailment-enriched	True	one_claim	binary	False	7	**23.4**	**4.6**	**13.3**	**84.0**
	entailment-only	True	one_premise	–	False	7	19.7	3.4	11.8	82.1
BERTSum	raw	–	–	–	–	8	23.6	4.5	**13.4**	**83.8**
	entailment-enriched	True	one_claim	binary	False	8	**23.8**	**4.7**	13.2	83.7
	entailment-only	True	no_filter	–	False	8	19.9	3.6	11.6	82.0
K-Means	raw	–	–	–	–	6	**22.2**	**4.1**	**11.9**	**83.4**
	entailment-enriched	True	one_claim	binary	–	6	20.7	3.6	11.5	82.8
	entailment-only	True	no_filter	–	–	6	16.5	2.6	10.4	80.7
DBSCAN	raw	–	–	–	–	10	**21.9**	**4.3**	**12.3**	**83.4**
	entailment-enriched	True	one_claim	binary	–	10	19.9	3.3	11.4	82.7
	entailment-only	True	no_filter	–	–	10	14.2	2.2	8.9	68.1
MatchSum	raw	–	–	–	–	7	23.7	4.3	13.1	83.8
	entailment-enriched	False	no_filter	keep	False	7	**24.2**	**4.6**	**13.2**	**84.0**
	entailment-only	True	no_filter	–	False	7	20.0	3.6	11.8	82.1

(b) Reddit domain

TextRank	raw	–	–	–	–	4	25.4	4.8	14.4	84.3
	entailment-enriched	True	one_claim	keep	False	4	**26.9**	**5.3**	**15.4**	**84.7**
	entailment-only	True	one_premise	–	False	4	15.9	2.8	9.7	61.9
LexRank	raw	–	–	–	–	4	25.4	5.2	14.9	84.6
	entailment-enriched	True	one_claim	keep	False	4	**27.0**	**5.6**	**15.4**	**84.8**
	entailment-only	True	one_premise	–	False	4	15.8	2.7	9.6	62.3
BERTSum	raw	–	–	–	–	5	26.2	4.8	14.5	84.4
	entailment-enriched	True	one_claim	keep	False	5	**27.1**	**5.1**	**14.8**	**84.5**
	entailment-only	True	one_premise	–	False	5	15.9	2.7	9.8	62.3
K-Means	raw	–	–	–	–	6	**26.2**	**5.1**	**14.1**	**84.4**
	entailment-enriched	True	one_p_c	keep	–	6	25.3	4.6	14.0	84.2
	entailment-only	True	no_filter	–	–	6	15.1	2.5	9.6	61.6
DBSCAN	raw	–	–	–	–	6	25.6	**5.3**	**14.2**	**84.4**
	entailment-enriched	True	one_claim	keep	–	6	**26.0**	4.7	14.0	84.2
	entailment-only	False	no_filter	–	–	6	6.3	1.0	3.7	24.7
MatchSum	raw	–	–	–	–	5	26.3	4.5	14.1	84.3
	entailment-enriched	True	one_claim	keep	False	5	**26.8**	**4.9**	**14.5**	**84.5**
	entailment-only	True	no_filter	–	False	5	15.9	2.7	9.6	62.3

(c) Stack domain

TextRank	raw	–	–	–	–	7	22.5	4.9	14.3	82.2
	entailment-enriched	True	one_claim	keep	False	7	**25.2**	**5.6**	**15.9**	**83.0**
	entailment-only	True	one_premise	–	False	7	7.8	1.5	5.2	34.2
LexRank	raw	–	–	–	–	7	21.7	4.3	13.3	82.4
	entailment-enriched	True	one_claim	binary	False	7	**25.5**	**5.8**	**14.7**	**83.1**
	entailment-only	True	no_filter	–	False	7	7.8	1.4	5.1	34.2
BERTSum	raw	–	–	–	–	6	25.9	5.9	16.0	83.2
	entailment-enriched	True	one_claim	binary	False	6	**27.5**	**6.4**	**16.5**	**83.7**
	entailment-only	True	one_p_c	–	True	6	7.8	1.4	5.2	34.2
K-Means	raw	–	–	–	–	6	24.3	4.6	14.3	82.7
	entailment-enriched	True	one_p_c	binary	–	6	**26.2**	**5.9**	**14.9**	**83.2**
	entailment-only	True	one_p_c	–	–	6	7.8	1.4	5.2	34.1
DBSCAN	raw	–	–	–	–	6	23.3	4.8	14.0	**82.6**
	entailment-enriched	True	one_claim	binary	–	6	**25.2**	**5.5**	**14.1**	81.5
	entailment-only	False	one_premise	–	–	6	1.9	0.3	1.2	7.2
MatchSum	raw	–	–	–	–	6	25.7	5.7	15.5	83.1
	entailment-enriched	True	one_claim	binary	False	6	**27.4**	**6.3**	**16.2**	**83.6**
	entailment-only	True	one_claim	–	True	6	7.9	1.4	5.2	34.2

(d) Email domain