Abstractive Text Summarization of Disaster-Related Documents

Nasik Muhammad Nafi*
Kansas State University
nnafi@ksu.edu

Avishek Bose*
Kansas State University
abose@ksu.edu

Sarthak Khanal*
Kansas State University
sarthakk@ksu.edu

Doina Caragea
Kansas State University
dcaragea@ksu.edu

William H. Hsu
Kansas State University
bhsu@ksu.edu

ABSTRACT

Abstractive summarization is intended to capture key information from the full text of documents. In the application domain of disaster and crisis event reporting, key information includes disaster effects, cause, and severity. While some researches regarding information extraction in the disaster domain have focused on keyphrase extraction from short disaster-related texts like tweets, there is hardly any work that attempts abstractive summarization of long disaster-related documents. Following the recent success of Reinforcement Learning (RL) in other domains, we leverage an RL-based state-of-the-art approach in abstractive summarization to summarize disaster-related documents. RL enables an agent to find an optimal policy by maximizing some reward. We design a novel hybrid reward metric for the disaster domain by combining Vector Similarity and Lexicon Matching (*VecLex*) to maximize the relevance of the abstract to the source document while focusing on disaster-related keywords. We evaluate the model on a disaster-related subset of a CNN/Daily Mail dataset consisting of 104,913 documents. The results show that our approach produces more informative summaries and achieves higher *VecLex* scores compared to the baseline.

Keywords

Disaster Reporting, Text Summarization, Information Extraction, Reinforcement Learning, Evaluation Metrics.

INTRODUCTION

Abstractive summarization aims to reduce human effort in reading documents by analyzing text and paraphrasing it in a concise format that preserves key topics and information vital to text understanding. Such natural language processing (NLP) tasks have been studied for a long time and applied to various purposes such as defense and surveillance activity monitoring (Ackerman and Miratrix 2013), natural disaster and crisis response management (L. Li and T. Li 2014), condensing journalistic text for socio-political events (Sethi et al. 2017), business, and economics (C. Wu and C. Liu 2003), etc.

A significant and time-critical use of summarization task is in the domain of crisis monitoring of disaster events, where critical information is embedded in data of high volume and velocity, including text communications. An informative summary of the data stream can help the first responders, and the emergency management teams to gain situational awareness and manage critical resources effectively (Zade et al. 2018). Moreover, such a technique can be implemented in real-time news feeds during crisis events to capture aggregated responses and dynamically track topics about the events.

[*] Authors contributed equally

The approaches used for text summarization can be broadly classified as extractive and abstractive (Munot and Govilkar 2014). Extractive approaches select existing words, phrases, or even whole sentences that are considered informative, from the source text. Although the extractive method captures key information from the text, it is prone to extracting explicitly mentioned keyphrases and sentences, and sometimes fails to induce the cognitive meaning of the provided source document. The problem becomes more complex if the length of the summary is limited in the number of sentences, and the extractive approach cannot serve the principal purpose of summarization. Moreover, the basic concept of extractive summarization is very different from how a human performs the same task (Verma and Lee 2017).

Abstractive summarization, however, is a technique of generating a summary by capturing the semantic representation of a given text. Abstractive summarization techniques, generally, use a generative approach (H. Lin and V. Ng 2019; L. Liu et al. 2017; C. Li et al. 2018) to generate sentences that may contain words not present in the source document. An abstractive summary, in that sense, is closer to what a human would generate (Zhang et al. 2019). Thus, abstractive summarization's sub-tasks include extracting nuanced semantic relations, capturing important syntactic information, and word sense inference, which together have shown to produce a condensed synopsis for sentences in a source text (Narayan et al. 2018).

Despite the benefits of abstractive approaches, extractive approaches are still considered state-of-the-art for summarization (Y. Wu and Hu 2018; Jadhav and Rajan 2018), due to their simplicity and better performance over basic abstractive approaches. However, often the extractive approaches fail to summarize certain key elements useful in a report, such as answers to *what, who, where, when, how*, etc. questions, if these questions are addressed across the whole source document, as opposed to a limited number of sentences. Nevertheless, the elements above make a significant impact in summarizing the reports in the domain of disaster, surveillance, and crisis management, and need to be concisely incorporated in summaries (Kropczynski et al. 2018). Therefore, choosing abstractive methods over extractive methods for summarizing reports relevant to disasters and crises would facilitate the generation of more informative summaries, not restricted to sentences that directly take words or word sequences from the source text.

In our work, we leverage an existing state-of-the-art approach for abstractive summarization (Chen and Bansal 2018), which uses RL to improve the summary of a single document. We adapt this approach to the disaster domain by proposing a new reward function based on sentence vector similarity and lexicon matching. We incorporate two common word embedding techniques for sentence vectorization, specifically (i) Word2Vec (Mikolov et al. 2013), and (ii) Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018). We use a predefined disaster/crisis lexicon (Olteanu et al. 2014) for lexicon matching.

Research works focusing on abstractive summarization generally use performance evaluation metrics designed for extractive approaches such as ROUGE score (C.-Y. Lin 2004) and Meteor (Banerjee and Lavie 2005). Metrics like n-gram based ROUGE scores used for the extractive summarization approaches are not very appropriate to use with abstractive summarization as the summaries may include words that do not directly appear in the original document. Some recent works (J.-P. Ng and Abrecht 2015; Sun and Nenkova 2019; Ray Chowdhury et al. 2019) have proposed word embedding based evaluation metrics. However, our analysis showed that higher embedding-based similarity doesn't always correspond to better summarization results. To overcome this limitation and to take advantage of disaster domain knowledge, we propose to use an evaluation metric that exactly matches the proposed reward function, which is based on vector similarity and disaster-related lexicon matching.

The use of the reward/evaluation metric that captures disaster lexical information and semantic information makes our approach very suitable for abstractive summarization tasks of long single-documents related to disaster and crisis, and thus enables the generation of informative synopses of disaster-related reports.

We have assembled a large dataset of disaster documents and corresponding summaries and conducted an evaluation of the abstractive summarization approach using both the traditional ROUGE score and our proposed evaluation metric. Experimental results indicate performance improvements over the baseline approach, which uses the ROUGE score as a reward function (Chen and Bansal 2018).

To summarize, our key contributions are as follows:

- We present a disaster and crisis-related data set consisting of 104,913 news reports (and their corresponding summaries) crawled using disaster-related keywords. We used this dataset to train and evaluate our models.
- We propose a reward function based on a vector embedding similarity score and a disaster-related lexicon matching score that we name *VecLex* reward. In addition, we propose a new training approach for summary generation, which uses the *VecLex* reward for the Reinforcement Learning (RL) agent training.

- We use an RL-based approach, with the proposed *VecLex* reward metric, to train models for abstractive summarization of long disaster and crisis-related text documents. Experimental results show the effectiveness of the reward metric in generating informative summaries for disaster documents.
- We conduct a comparative analysis of our proposed approach (which uses the *VecLex* reward) with the original summarization approach (which uses the ROUGE score for reward), and evaluate the results in terms of both ROUGE score and *VecLex* score.

RELATED WORK

Research on automatic summarization of text has been of interest for more than half a century (Luhn 1958). Considering the limitations of extractive summarization mentioned in the earlier section, research on abstractive approaches have been of interest lately. However, the generation of sentences from scratch as opposed to selection from a source is an inherently difficult task (Munot and Govilkar 2014). There have been many research works that attempt to use some combinations of both approaches to overcome their individual shortcomings (Yao et al. 2017).

The recent research on abstractive summarization has mostly been focused on the use of recurrent neural networks (RNN) (Kryscinski et al. 2018). A sequence-to-sequence model is one of the most researched neural network architectures used for summarization tasks (Kryscinski et al. 2018). Although RNNs were considered state-of-the-art approaches for NLP tasks in the last few years, more recent research has been focused on the application of the attention mechanism and transformers in text summarization (Ruder 2020). The use of pre-trained embeddings and more recently, Google BERT embeddings (Y. Liu and Lapata 2019) has produced significant improvements in abstractive summarization. Approaches based on reinforcement learning (Y. Wu and Hu 2018) and generative adversarial network (L. Liu et al. 2017) have also been used to improve the performance on this task.

The most popular metric for evaluation of text summaries, ROGUE (C.-Y. Lin 2004), uses an n-gram based matching or the longest sub-sequence based matching technique. Other metrics, such as *Density* and *Redundancy* (Fan et al. 2018), fail to capture the relevance of the generated summary and can not be used as a stand-alone metric. Another recently proposed metric proposed by Kryściński et al. (2019) attempts to check the factual consistency of the generated sentence with the source document. As a result, a sentence representing the correct information from the corresponding source document, but regarded as “unimportant” considering the essence of the source document, can be ranked higher. As these metrics are either inherently biased towards extractive summaries or designed to ensure certain properties, considering the associated bias and rigidity, there is a need for semantic similarity-based metrics that can cope with the flexibility required by abstractive summarization. Some of the available semantic-similarity-based approaches, like the one proposed by Silva et al. (2014), require extensive pre-processing and semantic graph building to come up with vector differences between words and to calculate the similarity between sentences. Thus, this approach introduces a big overhead if used in model training. Another work (Y. Li et al. 2004) uses an augmentation based on word-order, together with the n-gram matching technique, to calculate similarity.

The summarization task in the disaster domain has mostly been focused on extractive approaches (Zhou et al. 2014; Kedzie et al. 2015). Moreover, most of the summarization approaches in this domain are designed for collections of short texts, such as tweets (Rudra, Goyal, et al. 2018; Rudra, Sharma, et al. 2018). Despite the obvious usefulness in news portals and other interactive information retrieval systems available during disasters, there seems to be a lack of research in abstractive summarization techniques and evaluation metrics for long disaster-related documents.

METHODS

We leverage an existing model proposed by Chen and Bansal (2018), which has been shown to perform better than common baselines (See et al. 2017) in abstractive summarization tasks. We will use this approach as a base architecture (and a baseline) for our proposed variant.

Base Abstractive Sumarization Model

Figure 1 shows the architecture of the base model, which includes three sub-modules, Extractor, Abstractor and Reinforcement Learning Agent, as described below.

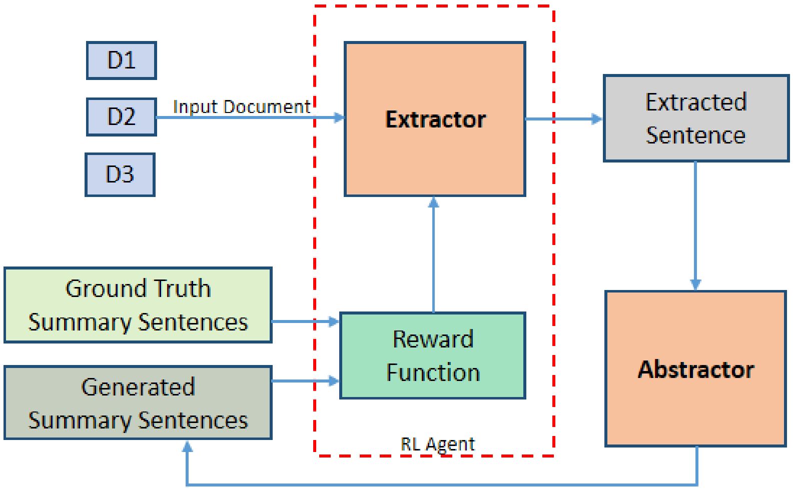

Figure 1. Graphical Representation of the Baseline Architecture

Extractor

The Extractor sub-module includes a Long Short Term Memory-Recurrent Neural Network (LSTM-RNN) and a Convolutional Neural Network (CNN) architecture. The temporal CNN generates a sentence encoding/representation for each sentence in the source document, while the Bidirectional LSTM-RNN aims to capture long-range semantic relationships between sentences considering the whole document. Another LSTM-RNN trains a 'Pointer Network' to extract salient sentences from the encoded sentence representations of the document. Here, the LSTM-RNN works as a decoder model that consists of a two-step attention process. The first step is required to get a context vector and the second step is for getting the extraction probabilities. It considers the sentences having high n-gram matching with the ground truth sentences as summary candidates. Thus, in each iteration, this model is able to select highly probable summary sentences.

Abstractor

The Abstractor sub-module models an abstraction function for the extracted summary sentences in a given document to modify and paraphrase an extracted sentence to a concise summary sentence. This sub-module consists of two different components named: (i) Sequence-Attention-Sequence module; and (ii) Copy mechanism module. Inside the Sequence-Attention-Sequence component, an encoder-aligner-decoder method (Bahdanau et al. 2014); (Luong et al. 2015) with bilinear multiplicative attention generates context vectors of provided sentences. The Copy mechanism, on the other hand, assists the decoder to predict an extended word vocabulary by determining copy probability based on some function parameters.

Reinforcement Learning (RL) Agent

This sub-module is based on a reinforcement learning approach that takes ROUGE score as a reward value to train and improve the extractor sub-module in an iterative fashion. This enables the extractor to generate summaries with higher ROUGE scores. ROUGE (C.-Y. Lin 2004) stands for "Recall-Oriented Understudy for Gisting Evaluation" and denotes a set of evaluation metrics used frequently in summarization and machine translation of text data. The set of metrics mentioned above includes ROUGE-N which considers n-gram matching, ROUGE-L which considers the Longest Common Subsequence, and ROUGE-S which considers Skip-Bigram based co-occurrence between generated text and source text.

Proposed Variant

To address the limitations of the conventional n-gram matching based approach used frequently with earlier abstractive approaches, we propose a new vector similarity-based training approach and a hybrid reward function.

Model Adaptation and Extractor Training

We adopted the base architecture described above, however, took a different training approach for the Extractor and the RL-Agent. We incorporated the vector similarity score in the basic Extractor training to make the entire candidate sentence selection process to be based on vector similarity. Here, for each ground truth summary sentences, we considered the most similar document sentence according to vector similarity scores as training labels (i.e., best contextually matched document sentence). In particular, these training labels facilitate the initial Maximum Likelihood (ML) based Extractor training. This ML based training helps to overcome the instability that arises in case of end-to-end training with a randomly initialized network. We calculated the vector similarity between two sentence vectors using cosine similarity. In our analysis, we used the following two popular vector embedding methods for sentence vectorization in order to measure the vector similarity of sentences:

- **Word2Vec-based Vector Similarity:** To calculate the word2vec (Mikolov et al. 2013) based vector similarity score, instead of using only the average for each dimension, we also take the minimum and maximum along each dimension of the word vectors over all the words in a summary. This leads to the generation of three vectors corresponding to min, max, and average, each having the same number of dimensions as the word vectors. We concatenate the vectors in the sequence of min, average, and max.

- **BERT-based Vector Similarity:** We also utilize the vector similarity score between the generated summary sentences and the original sentences from the corresponding vectors generated by the BERT pre-trained model. BERT is a method to pre-train language representation models, which produce high-quality sentence embedding vectors on a given text corpus. Unlike word embedding vectors such as Word2Vec (Mikolov et al. 2013) and Glove (Pennington et al. 2014), BERT can provide distinct vector embeddings for similar words written in different contexts in a sentence, because it simultaneously focuses on three specific vector building processes, characterized as token embedding, sentence embedding, and transformer positional embedding for each token written in a provided sentence. There are two BERT models: (i) BASE, and (ii) LARGE. We used the BASE model which has 12 layers of transformer encoders. In BERT, each output per token from each layer can be used as a word embedding. We choose the final layer as the output of the BERT model.

VecLex Reward Formulation and RL-Agent Training

In order to take advantage of domain knowledge, we integrate the disaster-related lexicon matching score in the reward function. We calculate the *VecLex* reward as a linear combination of the vector similarity score and lexicon matching score. We define the lexicon matching score as the ratio between the number of keywords matched with a lexicon and the total number of words in the generated summary. To calculate the vector similarity score, we transform the ground truth and the generated summary text to their vector representations using Word2Vec and BERT word embeddings and determine the cosine similarity between the vectors. Finally, this reward trains the RL-Agent to optimize the hybrid objective. Apart from the *VecLex* reward we also train the RL-Agent using only the vector similarity. We use a lexicon proposed by Olteanu et al. (2014) that contains 380 keywords found frequently in disaster-related reports. Incorporation of the lexicon matching score truncates the candidate sentence list given by the Extractor by putting more importance on the sentences which have disaster-related keywords. The number of sentences in a generated summary is decided dynamically for an input document. This is done as originally proposed by Chen and Bansal (2018) by including a STOP action to the policy action space. This enables the RL-Agent to learn when to stop generating summary sentences and to avoid extraneous information. Our proposed variant excludes the reranking strategy applied by Chen and Bansal (2018), as we observed that the summarization using a beam-search is much more expensive compared to the qualitative improvement of the generated summaries. Using the process described above, the abstractive summarization framework becomes semantically satisfiable and capable of generating contextually better disaster-related summaries.

Model Evaluation

We use Average *VecLex* score and Average Embedding Similarity (AES) score to measure the overall quality of the summaries, generated from the whole corpus, using vector similarity and lexicon matching. We calculate *VecLex* score as described in the previous paragraph. We use the Word2Vec-based vectorization approach as mentioned earlier to calculate the similarity between the ground truth and the corresponding generated summary sentences. Then, we take an average of the scores over all the documents of our corpus.

DATA PREPARATION

We extracted disaster-related documents from the CNN/Daily Mail dataset (Hermann et al. 2015) using a lexicon (Olteanu et al. 2014) related to disasters and crises by running a BASH script. The convention was that if we found at least three distinct tokens out of 380 tokens from the lexicon in any document of the data set, we would select that document. We collected 101,755 documents for training, 1,022 documents for validation, and 2,136 documents for testing our model. Each document is saved in JSON format and includes a long report of a disaster or crisis-related news event along with two to six lines of summary sentences.

EXPERIMENTAL SETUP

We evaluate the baseline model (Chen and Bansal 2018) in two ways. First, we use the pre-trained model (trained on the whole CNN/Daily Mail corpus) to generate summaries of the disaster-related documents. Then, we train the network only on the extracted disaster-related documents and generate summaries using that model.

We developed a Python script to calculate the *VecLex* and *AES* scores based on the Word2Vec vector similarity. We used the score calculated by the script to feed in the RL-based extractor sub-module of our chosen network. Also, we incorporated this script into multiple phases of the training pipeline to enable easy integration of vector similarity scores. The word embedding of pre-trained Word2Vec is available in 300 dimensions. As a result, the final vector representation of the summary text is 900 dimensional which is three times the original word vectors. Finally, we calculate the cosine similarity value between the two vectors produced from the given ground truth summary and the generated summary. In the case of BERT vectors, a few studies suggested that one of the best performing configurations is to sum the last 4 layers, although this depends on the domain. For our work, we empirically choose the final layer output because we got a relatively good result by using this layer. The dimension of the BERT sentence vector is 768.

For the LSTM module, we used 256 hidden state dimensions with only one layer and bidirectionality enabled. Adam was used as the optimizer, with a learning rate of 0.001 for the abstractor and extractor training, and 0.0001 for the RL-Agent training. In addition, we clipped the gradient norm larger than 2.0 and used a learning rate decay of 0.5.

In the end, we evaluated the performance of the proposed summarization approach on the test dataset. All experiments were run on the extracted disaster-related dataset described above. As discussed earlier, our selected code base for the baseline network (Chen and Bansal 2018) has an Extractor part that got trained with Maximum Likelihood estimation, followed by an RL-Agent based training. We experimented with incorporating vector similarity in different parts of the network. For both types of summary embeddings, we repeated the following five experiments:

1. Only vector similarity reward-based RL-Agent training.
2. *VecLex* reward-based RL-Agent training.
3. Vector similarity-based extractor training, followed by a vector similarity reward-based RL-Agent training.
4. Vector similarity-based extractor training, followed by a *VecLex* reward-based RL-Agent training.
5. Vector similarity-based extractor training, followed by a ROUGE score reward-based RL-Agent training.

EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we investigate the performance of vector similarity based approaches in disaster-related summarization tasks. Table 1 and Table 2 illustrate two examples of generated summaries along with the ground truth summary. Table 3 presents the overall performance of the trained models in terms of *AES* and *VecLex* scores, in addition to the ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. The detailed evaluation of ROUGE-1, ROUGE-2, ROUGE-L in terms of precision, recall, and F1 scores for every model, is summarized in Table 4. In analyzing and interpreting the experimental results, we answer several research questions, as follows.

- *Using the VecLex reward, is there any qualitative improvement in the generated summaries?*

  Relevance and readability are the two important features that need to be analyzed in qualitative analysis of abstractive summaries. The example in Table 1 talks about an avalanche in Mount Everest Base Camp area. The ground truth clearly indicates the location of the event, its cause, and its severity in terms of deaths, injuries, and people affected. The summary generated by the baseline misses two important pieces of information: (i) the exact location of Mount Everest; (ii) the number of people who were injured and died on the mountain. However, summaries generated by our approaches include the line "at least 18 people have died in an avalanche on mount everest" which is partially present in the ground truth and also captures the information missed by the baseline model.[1]

Table 1. Example 1 of generated summaries by comparison with the corresponding ground truth summary and the baseline summary

Ground Truth summary
powerful 7.8 magnitude earthquake caused an avalanche on mount everest . at least 18 people have died and more than 30 injured on the mountain . there are reports the avalanche has buried people in tents at base camp . the earthquake - nepal 's worst in 81 years - has killed at more than 1,300.
Summary generated by the baseline (Chen and Bansal 2018)
the avalanche buried part of base camp , raising fears for the safety of climbers . the avalanche was caused by a powerful 7.8 magnitude which struck nepal . a number of britons are among those who have not been heard from since the quake.the identities of those who died in the avalanche have not yet been released . at least 1,341 people have died as rescue teams continue to search for survivors .
Summary generated by the VecLex reward (Word2Vec-based)
the avalanche buried part of base camp , raising fears for the safety of hundreds of climbers . number of britons are among those who have not been heard from the avalanche . at least 18 people have died in an avalanche on mount everest . the avalanche was caused by a powerful 7.8 magnitude earthquake which struck nepal . at least 1,341 people have died as rescue teams continue to search for survivors .
Summary generated by the VecLex reward (BERT-based)
at least 18 people have died in an avalanche on mount everest . the avalanche buried part of base camp , raising fears for the safety of hundreds of climbers . number of britons are among those who have not been heard from the avalanche . the avalanche was caused by a powerful 7.8 magnitude earthquake which struck nepal . at least 1,341 people have died as rescue teams continue to search for survivors .

Table 2. Example 2 of generated summaries by comparison with the corresponding ground truth summary and the baseline summary

Ground Truth summary
4,000 on board the carnival spirit have been stranded outside sydney harbour . passengers were to disembark the ship on tuesday but the wild storm prevented it from entering the harbour . the vessel has suffered damage with smashed glass panels and a door ripped open by the crashing waves . carnival spirit was returning from a 12-night cruise to new caledonia , vanuatu and fiji .
Summary generated by the baseline (Chen and Bansal 2018)
passengers on board a cruise ship have finally arrived at sydney harbour . huge waves as high as 40 feet smashed glass panels , shattered plates and cups . the ship was carrying 2,500 holidaymakers and 1,500 crew . holidaymakers were due to disembark the carnival spirit on tuesday morning . passengers have been ' vomiting for two days ' due to rough seas conditions .
Summary generated by the VecLex reward (Word2Vec-based)
cruise ship passengers on board carnival spirit as sydney was battered by storm . huge waves as high as 40 feet smashed glass panels , shattered plates and cups . pilot finally boarded this morning and the ship , carrying 2,500 holidaymakers and 1,500 crew, docked at around 10 am . the 2500 passengers on board a cruise ship has been stranded at sea . holidaymakers were due to disembark the carnival spirit on tuesday morning .
Summary generated by the VecLex reward (BERT-based)
cruise ship passengers on board carnival spirit as sydney was battered by storm . huge waves as high as 40 feet smashed glass panels , shattered plates and cups . pilot finally boarded this morning and the ship , carrying 2,500 holidaymakers and 1,500 crew , docked at around 10 am . holidaymakers were due to disembark the carnival spirit on tuesday morning . the 2500 passengers on board a cruise ship has been stranded at sea .

The example in Table 2 describes the incident of a storm in the ocean, and its consequences faced by a cruise ship. For this example, the output generated by the baseline misses an aspect that is very crucial

[1] In the tables, sentences present in the ground truth and in our generated summary, but not in the baseline summary, are highlighted with red. New important pieces of information captured only by our summary are highlighted with blue.

considering disaster-related events. It fails to indicate the type of disaster. On the contrary, our approach mentions at the beginning that the incident is caused by a storm. This also increases the coherence in the sentence sequence, and, in turn, increases readability, as well as semantic understanding. Additionally, our generated summary captures the time when the ship docked. In the tables, the lines which are present in the ground truth and in our generated summary, but not in the baseline summary, are marked red. New important pieces of information captured only by our summaries are marked in blue color.

Table 3. Comparison of different models using different evaluation metrics

Model	***AES***	***VecLex***	***ROUGE-1***	***ROUGE-2***	***ROUGE-L***
Chen & Bansal, 2018 (Pretrained Model)	95.518	71.426	**39.865**	**17.371**	**37.252**
Chen & Bansal, 2018 (Trained only on disaster data set)	95.471	71.433	39.555	16.976	36.933
Word2Vec-based					
1. Only Vector Similarity reward	95.499	70.162	37.106	15.125	34.665
2. VecLex - Vector similarity + Disaster lexicon matching reward	95.410	**71.585**	39.171	16.708	36.568
3. Vector similarity-based Extractor + vector similarity reward	**95.778**	68.583	36.788	15.750	34.547
4. Vector similarity-based Extractor + VecLex Reward	95.557	71.386	39.498	16.862	36.849
5. Vector similarity-based Extractor + ROUGE Reward	95.543	70.516	39.506	16.919	36.860
BERT-based					
1. Only Vector Similarity reward	95.644	70.116	39.111	16.793	36.588
2. VecLex - Vector Similarity + Disaster lexicon matching reward	95.492	71.506	39.321	16.665	36.713
3. Vector similarity-based Extractor + vector similarity reward	95.720	69.666	38.577	16.612	36.155
4. Vector similarity-based Extractor + VecLex Reward	95.393	71.085	38.663	16.263	36.039
5. Vector similarity-based Extractor + ROUGE Reward	95.560	70.875	39.366	16.894	36.811

- *Does the use of the VecLex reward improve the performance of the model? Do summaries with better VecLex scores correspond to qualitatively better summaries?*

The use of the lexicon matching score in association with vector similarity improves the performance for both types of models. As can be seen in Table 3, for both Word2Vec-based and BERT-based embeddings, the best performing model is the one when the RL-Agent is trained with the *VecLex* reward. This variant clearly outperforms the baseline. The results show that the addition of the lexicon matching score helps to discard unnecessary sentences that are not related to various kind disasters, while the vector similarity helps capture the main essence of the summary. In the examples shown in Table 1 and Table 2, the summaries generated by *VecLex* reward include sentences that are not captured by the baseline, and in those sentences there are the words "died" and "storm", respectively. These two words are also present in our used lexicon. This reveals the potential influence of the lexicon matching score and of the whole *VecLex* reward. Thus, we can conclude that summaries with better *VecLex* scores correspond to qualitatively better summaries in case of disaster and crisis-related documents.

- *What are the insights of using only vector similarity-based training? Can AES be a better metric for summarization?*

For both Word2Vec and BERT based embeddings, the models trained with vector similarity-based extractor and vector similarity-based reward generated higher *AES* score than others. However, a detailed investigation reveals that both of those models generated longer summaries and thus increased the *AES* score by keeping more sentences. This is also evident from the very high recall score produced by the vector-based reward and training approaches presented in Table 4. This leads to the idea that we need another measurement, which will either enable the model to discard some sentences or penalize the model when the summary is too long. While *AES* score cannot be a standalone metric for final evaluation, it can be a robust metric when accompanied by other metrics having certain characteristics such as the *VecLex* metric.

Table 4. Analysis of ROUGE scores for models trained using vector similarity based approach

Model		***ROUGE-1***	***ROUGE-2***	***ROUGE-L***
Word2Vec-based				
Only vector similarity as reward	Recall	49.997	20.416	46.737
	Precision	31.389	12.812	29.321
	F1 Score	37.106	15.125	34.665
Vector similarity with Lexicon matching reward	Recall	45.289	19.444	42.301
	Precision	36.595	15.558	34.156
	F1 Score	39.171	16.708	36.568
Vector similarity-based Extractor + Vector Similarity Reward	Recall	56.810	24.625	53.411
	Precision	28.561	12.171	26.811
	F1 Score	36.788	15.750	34.547
Vector similarity-based Extractor + Vector similarity with Lexicon matching reward	Recall	47.873	20.630	44.676
	Precision	35.424	15.036	33.046
	F1 Score	39.498	16.862	36.849
Vector similarity-based Extractor + ROUGE Reward	Recall	48.157	20.770	44.945
	Precision	35.424	15.124	33.051
	F1 Score	39.506	16.919	36.860
BERT-based				
Only vector similarity as reward	Recall	50.490	21.846	47.253
	Precision	33.591	14.368	31.419
	F1 Score	39.111	16.793	36.588
Vector similarity with Lexicon matching reward	Recall	47.102	20.090	43.975
	Precision	35.627	15.052	33.269
	F1 Score	39.321	16.665	36.713
Vector similarity-based Extractor + Vector Similarity Reward	Recall	52.711	22.946	49.431
	Precision	31.975	13.694	29.960
	F1 Score	38.577	16.612	36.155
Vector similarity-based Extractor + Vector similarity with Lexicon matching reward	Recall	45.355	19.265	42.302
	Precision	35.911	15.018	33.464
	F1 Score	38.663	16.263	36.039
Vector similarity-based Extractor + ROUGE Reward	Recall	48.681	21.095	45.546
	Precision	34.850	14.876	32.577
	F1 Score	39.366	16.894	36.811

- *Are the results of ROUGE scores, which are essentially based on n-gram matching, still comparable to the baseline results even though the models were trained using the vector similarity-based approach?*

Our results show that using a vector similarity score as a reward in the RL-Agent training does not produce better results in terms of the ROUGE scores. Moreover, only the vector similarity-based Maximum Likelihood estimation in association with RL reward makes the scenario worse. This may be due to the fact that the vector similarity-based approach tries to keep more sentences that have similar contexts. Consequently, this leads to higher recall but lower precision value resulting in a lower overall F1 score, as presented in Table 4. However, the models become more robust when the disaster-related lexicon matching score is incorporated in the reward function in addition to vector similarity. This helps to improve the F1 score by balancing the recall value and also leads to a better ROUGE score. We also experimented with the ROUGE score based reward in conjunction with the vector similarity-based extractor training.

CONCLUSIONS AND FUTURE WORK

We leveraged an existing abstractive summarization approach to generate summaries for long documents about disaster and crisis events. The abstractive summarization approach provides qualitatively better summaries when used with crisis-related documents, as seen in the results. Abstractive summaries should be able to answer relevant and key questions of interest. A step away from the metrics inherently biased to the extractive approach is a step towards better abstractive models. Taking this into consideration, we incorporated vector-based similarity and lexicon filtering into different sections of the base architecture, and we were able to achieve better disaster relevant summaries. One interesting thing to note is that with the vector similarity-based extractor and vector similarity reward, we get a significantly higher recall because the model tends to select more sentences from the document as candidates for the summary content. The use of lexicon-based filtering, however, lessens this effect and also makes the summaries more pertinent towards disaster at the same time. We are able to generate summaries that answer key questions of *who, what, where, when*, etc. that are significant in crisis documents.

Considering the recent successes in the NLP domain with the introduction of the transformers and their pre-trained models, abstractive models can become dominant in the crisis domain. Although we get good quality summaries with high *VecLex* values, the level of abstraction in the summaries produced can be improved further. As part of future work, we plan to fine-tune other state-of-the-art models with the *VecLex* reward using our disaster-related corpus. We also plan to incorporate location primitives into the reward function to further improve the the detection and usefulness of the w-questions mentioned above. Finally, it is also of interest to analyze how consistent our generated summaries are with respect to the factual information of a report.

ACKNOWLEDGEMENTS

We thank the National Science Foundation and Amazon Web Services for support from grant IIS-1741345, which enabled the research and the computation in this study. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either express or implied, of the National Science Foundation or Amazon Web Services.

REFERENCES

Ackerman, R. and Miratrix, L. (Nov. 2013). “Using text summarization for surveillance: A case study involving OSHA fatality and catastrophe reports”. In:

Bahdanau, D., Cho, K., and Bengio, Y. (2014). “Neural machine translation by jointly learning to align and translate”. In:

Banerjee, S. and Lavie, A. (2005). “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments”. In: pp. 65–72.

Chen, Y. and Bansal, M. (2018). “Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting”. In: *CoRR* abs/1805.11080. arXiv: 1805.11080.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805. arXiv: 1810.04805.

Fan, L., Yu, D., and Wang, L. (2018). “Robust Neural Abstractive Summarization Systems and Evaluation against Adversarial Information”. In: *ArXiv* abs/1810.06065.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). “Teaching Machines to Read and Comprehend”. In: *CoRR* abs/1506.03340. arXiv: 1506.03340.

Jadhav, A. and Rajan, V. (July 2018). “Extractive Summarization with SWAP-NET: Sentences and Words from Alternating Pointer Networks”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 142–151.

Kedzie, C., McKeown, K., and Diaz, F. (July 2015). “Predicting Salient Updates for Disaster Summarization”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 1608–1617.

Kropczynski, J., Grace, R., Coche, J., Halse, S., Obeysekare, E., Montarnal, A., Benaben, F., and Tapia, A. (2018). “Identifying actionable information on social media for emergency dispatch”. In: *Proceedings of the ISCRAM Asia Pacific*.

Kryściński, W., McCann, B., Xiong, C., and Socher, R. (2019). *Evaluating the Factual Consistency of Abstractive Text Summarization*. arXiv: 1910.12840 [cs.CL].

Kryscinski, W., Paulus, R., Xiong, C., and Socher, R. (2018). "Improving Abstraction in Text Summarization". In: *CoRR* abs/1808.07913. arXiv: 1808.07913.

Li, C., Xu, W., Li, S., and Gao, S. (June 2018). "Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 55–60.

Li, L. and Li, T. (Feb. 2014). "An Empirical Study of Ontology-Based Multi-Document Summarization in Disaster Management". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.2, pp. 162–171.

Li, Y., Bandar, Z., McLean, D., and O'Shea, J. (Jan. 2004). "A Method for Measuring Sentence Similarity and its Application to Conversational Agents." In:

Lin, C.-Y. (July 2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81.

Lin, H. and Ng, V. (2019). "Abstractive Summarization: A Survey of the State of the Art". In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 9815–9822.

Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., and Li, H. (2017). "Generative Adversarial Network for Abstractive Text Summarization". In: *CoRR* abs/1711.09357. arXiv: 1711.09357.

Liu, Y. and Lapata, M. (Nov. 2019). "Text Summarization with Pretrained Encoders". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 3728–3738.

Luhn, H. P. (Apr. 1958). "The Automatic Creation of Literature Abstracts". In: *IBM J. Res. Dev.* 2.2, pp. 159–165.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). "Effective approaches to attention-based neural machine translation". In:

Mikolov, T., Corrado, G., Chen, K., and Dean, J. (Jan. 2013). "Efficient Estimation of Word Representations in Vector Space". In: pp. 1–12.

Munot, N. and Govilkar, S. (Sept. 2014). "Comparative Study of Text Summarization Methods". In: *International Journal of Computer Applications* 102, pp. 33–37.

Narayan, S., Cohen, S. B., and Lapata, M. (Oct. 2018). "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1797–1807.

Ng, J.-P. and Abrecht, V. (Sept. 2015). "Better Summarization Evaluation with Word Embeddings for ROUGE". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1925–1930.

Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. (2014). "Crisislex: A lexicon for collecting and filtering microblogged communications in crises". In: *Eighth international AAAI conference on weblogs and social media*.

Pennington, J., Socher, R., and Manning, C. (Oct. 2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543.

Ray Chowdhury, J., Caragea, C., and Caragea, D. (2019). "Keyphrase Extraction from Disaster-related Tweets". In: *The World Wide Web Conference on - WWW '19*.

Ruder, S. (2020). *Tracking Progress in Natural Language Processing*.

Rudra, K., Goyal, P., Ganguly, N., Mitra, P., and Imran, M. (2018). "Identifying Sub-Events and Summarizing Disaster-Related Information from Microblogs". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: Association for Computing Machinery, pp. 265–274.

Rudra, K., Sharma, A., Ganguly, N., and Imran, M. (2018). "Classifying and summarizing information from microblogs during epidemics". In: *Information Systems Frontiers* 20.5, pp. 933–948.

See, A., Liu, P. J., and Manning, C. D. (2017). "Get to the point: Summarization with pointer-generator networks". In:

Sethi, P., Sonawane, S., Khanwalker, S., and Keskar, R. B. (Dec. 2017). "Automatic text summarization of news articles". In: *2017 International Conference on Big Data, IoT and Data Science (BID)*, pp. 23–29.

Silva, N. de, Silva, W., Gunasinghe, U., Perera, A., Sashika, W., and Premasiri, W. (Dec. 2014). "Sentence Similarity Measuring by Vector Space Model". In:

Sun, S. and Nenkova, A. (Nov. 2019). "The Feasibility of Embedding Based Automatic Evaluation for Single Document Summarization". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 1216–1221.

Verma, R. M. and Lee, D. (2017). "Extractive Summarization: Limits, Compression, Generalized Model and Heuristics". In: *CoRR* abs/1704.05550. arXiv: 1704.05550.

Wu, C. and Liu, C. (Jan. 2003). "Ontology-based Text Summarization for Business News Articles." In: pp. 389–392.

Wu, Y. and Hu, B. (2018). "Learning to Extract Coherent Summary via Deep Reinforcement Learning". In: *CoRR* abs/1804.07036. arXiv: 1804.07036.

Yao, J.-G., Wan, X., and Xiao, J. (Nov. 2017). "Recent Advances in Document Summarization". In: *Knowl. Inf. Syst.* 53.2, pp. 297–336.

Zade, H., Shah, K., Rangarajan, V., Kshirsagar, P., Imran, M., and Starbird, K. (2018). "From situational awareness to actionability: Towards improving the utility of social media data for crisis response". In: *Proceedings of the ACM on human-computer interaction* 2.CSCW, pp. 1–18.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. arXiv: 1912.08777 [cs.CL].

Zhou, W., Shen, C., Li, T., Chen, S., and Xie, N. (Aug. 2014). "Generating textual storyline to improve situation awareness in disaster management". In: *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pp. 585–592.