Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference

Toward Extractive Summarization of Online Forum Discussions via Hierarchical Attention Networks

Sansiri Tarnpradab, Fei Liu, Kien A. Hua
Department of Computer Science
University of Central Florida, Orlando, FL 32816
sansiri@knights.ucf.edu {feiliu, kienhua}@cs.ucf.edu

Abstract

Forum threads are lengthy and rich in content. Concise thread summaries will benefit both newcomers seeking information and those who participate in the discussion. Few studies, however, have examined the task of forum thread summarization. In this work we make the first attempt to adapt the hierarchical attention networks for thread summarization. The model draws on the recent development of neural attention mechanisms to build sentence and thread representations and use them for summarization. Our results indicate that the proposed approach can outperform a range of competitive baselines. Further, a redundancy removal step is crucial for achieving outstanding results.

Introduction

Online forums play an important role in shaping public opinions on a number of issues, ranging from popular tourist destinations to major political events. As a form of new media, the influence of forums is on the rise and rivals that of traditional media outlets (Stephen and Galak 2012). A forum thread is typically initiated by a user posting a question or comment through the website. Others reply with clarification questions, further details, solutions, and positive/negative feedback (Bhatia, Biyani, and Mitra 2014). This corresponds to a community-based knowledge creation process where knowledge of enduring value is preserved (Anderson et al. 2012). It is not uncommon that forum threads are lengthy and comprehensive, containing hundreds of pages of discussion. In this work we seek to generate concise forum thread summaries that will benefit both the newcomers seeking information and those who participate in the discussion.

Few studies have examined the task of forum thread summarization. Traditional approaches are largely based on multi-document summarization frameworks. Ding and Jiang (2015) presented a preliminary study on extracting opinionated summaries for online forum threads. They analyzed the discriminative power of a range of sentence-level features, including relevance, text quality and subjectivity. Bhatia et al. (2014) studied the effect of dialog act labels on predicting summary posts. They define a thread summary as a collection of relevant posts from a discussion. Ren et al. (2011) approached the problem using hierarchical Bayesian models and performed random walks on the graph to select summary sentences. The aforementioned studies used datasets ranging from 10 to 400 threads. Due to the lack of annotated datasets, supervised summarization approaches have largely been absent from this space.

In this work we introduce a novel supervised thread summarization approach that is adapted from the hierarchical attention networks (HAN) proposed in (Yang et al. 2016). The model draws on the recent development of neural attention mechanisms. It learns effective sentence representation by attending to important words, and similarly learns thread representation by attending to important sentences in the thread. Hierarchical network structures have seen success in both document modeling (Li, Luong, and Jurafsky 2015) and machine comprehension (Yin, Ebert, and Schutze 2016). To the best of our knowledge, this work is the first attempt to adapt it to forum thread summarization. We further created a dataset by manually annotating 600 threads with human summaries. The annotated data allow the development of a supervised system trained in an end-to-end fashion. We compare the proposed approach against state-of-the-art summarization baselines. Our results indicate that the HAN models are effective in predicting summary sentences. Further, a redundancy removal step is crucial for achieving outstanding results.

Our Approach

We formulate thread summarization as a task that extracts relevant sentences from a discussion. A sentence is used as the extraction unit due to its succinctness. The task naturally lends itself to a supervised learning framework. Let $\boldsymbol{s}$=$[s_1, \cdots, s_N]$ be the sentences in a thread and $\boldsymbol{t}$=$[t_1, \cdots, t_N]$ be the binary labels, where 1 indicates the sentence is in the summary and 0 otherwise. The task of forum thread summarization is to find the most probable tag sequence given the thread sentences:

$$\arg\max_{\boldsymbol{t} \in \mathcal{T}} p(\boldsymbol{t}|\boldsymbol{s}) \tag{1}$$

where $\mathcal{T}$ is the set of all possible tag sequences. In this work we make independent tagging decisions, where $p(\boldsymbol{t}|\boldsymbol{s}) = \prod_{i=1}^{N} p(t_i|\boldsymbol{s})$. We begin by describing the hierarchical attention networks (HAN; Yang et al., 2016) that are used to construct sentence and thread representations, followed by our adaptation of the HAN models to thread summarization. Below we use bold letters to represent vectors and matrices (e.g., $\boldsymbol{h}_t, \boldsymbol{W}$). Words and sentences are denoted by their indices.

Sentence Encoder. It reads an input sentence and outputs a sentence vector. Inspired by recent results in (Bahdanau, Cho, and Bengio 2015; Chen, Bolton, and Manning 2016), we use a bi-directional recurrent neural network as the sentence encoder. The model additionally employs an attention mechanism that learns to attend to important words in the sentence while generating the sentence vector.

Let $\boldsymbol{s}_i$=$[x_1, \cdots, x_T]$ be the i-th sentence and the words are indexed by t. Each word is replaced by a pretrained word embedding before it is fed to the neural network. We use the 300-dimension word2vec embeddings (Mikolov et al. 2013) pretrained on Google News dataset with about 100 billion words. While both gated recurrent units (GRU, Chung et al., 2014) and long short-term memory (LSTM, ochreiter and Schmidhuber 1997) are variants of recurrent neural networks, we opt for LSTM in this study due to its proven effectiveness in previous studies.

LSTM embeds each word into a hidden representation $\boldsymbol{h}_t$=LSTM$(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$. It employs three gating functions (input gate $\boldsymbol{i}_t$ (Eq.(2)), forget gate $\boldsymbol{f}_t$ (Eq.(3)), and output gate $\boldsymbol{o}_t$ (Eq.(4))) to control how much information comes from the previous time step, and how much will flow to the next. The gating mechanism is expected to keep information flow for a long period of time. In particular, Eq.(6) calculates the cell state $\boldsymbol{C}_t$ by selectively inheriting information from $\tilde{\boldsymbol{C}}_t$ (via the input gate) and from $\boldsymbol{C}_{t-1}$ (via the forget gate). Eq.(7) generates the hidden state by applying the output gate to $\tanh(\boldsymbol{C}_t)$. The equations are described below.

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}^i \boldsymbol{x}_t + \boldsymbol{U}^i \boldsymbol{h}_{t-1} + \boldsymbol{b}^i) \quad (2)$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}^f \boldsymbol{x}_t + \boldsymbol{U}^f \boldsymbol{h}_{t-1} + \boldsymbol{b}^f) \quad (3)$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}^o \boldsymbol{x}_t + \boldsymbol{U}^o \boldsymbol{h}_{t-1} + \boldsymbol{b}^o) \quad (4)$$

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}^c \boldsymbol{x}_t + \boldsymbol{U}^c \boldsymbol{h}_{t-1} + \boldsymbol{b}^c) \quad (5)$$

$$\boldsymbol{C}_t = \boldsymbol{i}_t \odot \tilde{\boldsymbol{C}}_t + \boldsymbol{f}_t \odot \boldsymbol{C}_{t-1} \quad (6)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{C}_t) \quad (7)$$

where $\odot$ is the element-wise product of two vectors. We additionally employ a bi-directional LSTM model that includes a forward-pass (Eq.(8)) and a backward pass (Eq.(9)). $\overrightarrow{\boldsymbol{h}_t}$ is expected to carry over semantic information from beginning of the sentence to the current time step; whereas $\overleftarrow{\boldsymbol{h}_t}$ encodes information from the current time step to the end of sentence. Concatenating the two vectors $\boldsymbol{h}_t$=$[\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}]$ produces a word representation that encodes the sentence-level context.

$$\overrightarrow{\boldsymbol{h}_t} = \text{LSTM}_1(\overrightarrow{\boldsymbol{h}_{t-1}}, \boldsymbol{x}_t) \quad (8)$$

$$\overleftarrow{\boldsymbol{h}_t} = \text{LSTM}_2(\overleftarrow{\boldsymbol{h}_{t-1}}, \boldsymbol{x}_t) \quad (9)$$

Next we describe the attention mechanism. Of key importance is the introduction of a vector $\boldsymbol{u}_w$ for all words, which is *trainable* and expected to capture "global" word saliency. We first project $\boldsymbol{h}_t$ to a transformed space and generates $\boldsymbol{u}_t$ (Eq.(10)). The inner product $\boldsymbol{u}_t^T \boldsymbol{u}_w$ is expected to signal the importance of the t-th word. It is converted to a normalized weight α_t through a softmax function (Eq.(11)).

$$\boldsymbol{u}_t = \tanh(\boldsymbol{W}^a \boldsymbol{h}_t + \boldsymbol{b}^a) \quad (10)$$

$$\alpha_t = \frac{\exp(\boldsymbol{u}_t^T \boldsymbol{u}_w)}{\sum_t \exp(\boldsymbol{u}_t^T \boldsymbol{u}_w)} \quad (11)$$

The sentence vector $\boldsymbol{s}_i$ is generated as a weighted sum of word representations, where α_t is a scalar value indicating the word importance (Eq.(12)).

$$\boldsymbol{s}_i = \sum_t \alpha_t \boldsymbol{h}_t \quad (12)$$

Thread Encoder. It takes as input a sequence of sentence vectors $\boldsymbol{s}$=$[\boldsymbol{s}_1, \cdots, \boldsymbol{s}_N]$ encoded using the sentence encoder described above and outputs a thread vector. Assume the sentences are indexed by i. The thread encoder employs the same network architecture as the sentence encoder. We summarize the equations below. Note that the attention mechanism additionally introduces a vector $\boldsymbol{u}_s$ for all sentences, which is *trainable* and encodes salient sentence-level content. The thread vector $\boldsymbol{s}$ is a weighted sum of sentence vectors, where α_i is a scalar value indicating the importance of the i-th sentence.

$$\overrightarrow{\boldsymbol{h}_i} = \text{LSTM}_3(\overrightarrow{\boldsymbol{h}_{i-1}}, \boldsymbol{s}_i) \quad (13)$$

$$\overleftarrow{\boldsymbol{h}_i} = \text{LSTM}_4(\overleftarrow{\boldsymbol{h}_{i-1}}, \boldsymbol{s}_i) \quad (14)$$

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}_i}, \overleftarrow{\boldsymbol{h}_i}] \quad (15)$$

$$\boldsymbol{u}_i = \tanh(\boldsymbol{W}^b \boldsymbol{h}_i + \boldsymbol{b}^b) \quad (16)$$

$$\alpha_i = \frac{\exp(\boldsymbol{u}_i^T \boldsymbol{u}_s)}{\sum_i \exp(\boldsymbol{u}_i^T \boldsymbol{u}_s)} \quad (17)$$

$$\boldsymbol{s} = \sum_i \alpha_i \boldsymbol{h}_i$$

Output Layer. Each sentence of the thread is represented using a concatenation of the corresponding sentence and thread vectors. Thus, both sentence- and thread-level context are taken into consideration when predicting if the sentence is in the summary. We use a dense layer and a cross-entropy loss for the output.

Two additional improvements are crucial for the HAN models: 1) ***pretrain***. The models are initially designed for text classification. Using the thread vectors and thread category labels (Bhatia, Biyani, and Mitra 2016), we are able to pretrain the HAN models on a text classification task. We hypothesize that the pretrained sentence and thread encoders are well-suited for the summarization task. 2) ***redundancy removal***. Supervised summarization models do not handle redundancy well. Following (Cao et al. 2017), we apply a redundancy removal step, where sentences of high relevance are iteratively added to the summary and a sentence is added if it contains at least 50% new bigrams that are not previously contained in the summary.

Data

Having described the HAN models for summarization in the previous section, we next present our data. We use forum threads collected by Bhatia et al. (2014) from tripadvisor.com and ubuntuforums.org. The data contain respectively 83,075 and 113,277 threads from TripAdvisor and Ubuntu-Forums. Among them, 1,480 and 1,174 threads have category labels (Bhatia, Biyani, and Mitra 2016) and are used for model pretraining. Bhatia et al. (2014) annotated 100 TripAdvisor threads with human summaries. In this work we extend the summary annotation with 600 more threads, making a total of 700 threads.[1] We recruited six annotators and instructed them to read each thread and produce a summary of 10% to 25% of the original thread length. They can use sentences in the thread or their own words. Two human summaries are created per thread. We set aside 100 threads as a dev set and report results on the rest 600 threads. In total, there are 34,033 sentences in the 600 threads. A thread contains 10.5 posts and 56.2 sentences averagely.

Further, we need to obtain sentence-level summary labels, where 1 means the sentence is in the gold-standard summary and 0 otherwise. This is accomplished using an iterative greedy selection process. Starting from an empty set, we add one sentence to the summary in each iteration such that the sentence produces the most improvement on ROUGE-1 scores (Lin 2004). The process stops if none could improve the ROUGE-1 scores, or if the summary has reached a pre-specified length limit of 20% of the total words in the thread. Note that, since there are two human summaries for every forum thread, ROUGE-1 scores measure the unigram overlap between the selected sentences and both of the human summaries. ROUGE 2.0 Java package was used for evaluation.

Experimental Setup

Unsupervised baselines. Our proposed approach is compared against a range of unsupervised baselines, including 1) ILP (Berg-Kirkpatrick, Gillick, and Klein 2011), a baseline integer linear programming (ILP) framework implemented by (Boudin, Mougard, and Favre 2015); 2) SUMBASIC (Vanderwende et al. 2007), an approach that assumes words occurring frequently in a document cluster have a higher chance of being included in the summary; 3) KL-SUM, a method that adds sentences to the summary so long as it decreases the KL Divergence; 4) LEXRANK (Erkan and Radev 2004), a graph-based summarization approach based on eigenvector centrality; 5) MEAD (Radev et al. 2004), a centroid-based summarization system that scores sentences based on length, centroid, and position.

Supervised baselines. We implemented two supervised baselines that use SVM and logistic regression to predict if a sentence is in the summary. We use the LIBLINEAR implementation (Fan et al. 2008) where features include 1) cosine similarity of current sentence to the thread centroid, 2) relative sentence position within thread, 3) number of words in the sentence excluding stopwords, 4) max/avg/total TF-IDF scores of the consisting words. The features are designed such that they carry similar information as achievable by the HAN models. We use the 100-thread dev set for tuning hyperparameters. The optimal ones are '-c 0.1 -w1 5' for LogReg and '-c 10 -w1 5' for SVM.

HAN configurations. The HAN models use RMSProp (Tieleman and Hinton 2012) for parameter optimization, which has been shown to converge fast in sequence learning tasks. The number of sentences per thread is set to 144 and number of words per sentence is 40. We produce 200-dimension sentence vectors and 100-dimension thread vectors. Dropout for word embeddings was 20% and the output layer 50%.

Evaluation metrics. ROUGE (Lin 2004) measures the n-gram overlap between system and human summaries. In this work we report ROUGE-1 and ROUGE-2 scores since these are metrics commonly used in the DUC and TAC competitions (Dang and Owczarzak 2008). Additionally, we calculate the sentence-level precision, recall, and f-scores by comparing system prediction with gold-standard sentence labels. All system summaries use a length threshold of 20% thread words.

Results

The experimental results of all models are shown in Table 1. The HAN models are compared with a set of unsupervised (ILP, Sum-Basic, KL-Sum, LexRank, and MEAD) and supervised (SVM, LogReg) approaches. We describe the observations below.

- First, HAN models appear to be more appealing than SVM and LogReg because there is less variation in program implementation, hence less effort is required to reproduce the results. HAN models outperform both LogReg and SVM using the current set of features. They yield higher precision scores than traditional models.
- With respect to ROUGE scores, the HAN models outperform all supervised and unsupervised baselines except MEAD. MEAD has been shown to perform well in previous studies (Luo et al. 2016) and it appears to handle redundancy removal exceptionally well. The HAN models outperform MEAD in terms of sentence prediction.
- Pretraining the HAN models, although intuitively promising, yields only comparable results with those without. We suspect that there are not enough data to pretrain the models and that the thread classification task used to pretrain the HAN models may not be sophisticated enough to learn effective thread vectors.
- We observe that the redundancy removal step is crucial for the HAN models to achieve outstanding results. It helps improve the recall scores of both ROUGE and sentence prediction. When redundancy removal was applied to LogReg, it produces only marginal improvement. This suggests that future work may need to consider principled ways of redundancy removal.

Related Work

There has been some related work on email thread summarization (Rambow et al. 2004; Wan and McKeown 2004;

[1]The data is available at http://tinyurl.com/jcqgcu8

System	ROUGE-1 R (%)	ROUGE-1 P (%)	ROUGE-1 F (%)	ROUGE-2 R (%)	ROUGE-2 P (%)	ROUGE-2 F (%)	Sentence-Level R (%)	Sentence-Level P (%)	Sentence-Level F (%)
ILP	24.5	41.1	29.3±0.5	7.9	15.0	9.9±0.5	13.6	22.6	15.6±0.4
Sum-Basic	28.4	44.4	33.1±0.5	8.5	15.6	10.4±0.4	14.7	22.9	16.7±0.5
KL-Sum	39.5	34.6	35.5±0.5	13.0	12.7	12.3±0.5	15.2	21.1	16.3±0.5
LexRank	42.1	39.5	38.7±0.5	14.7	15.3	14.2±0.5	14.3	21.5	16.0±0.5
MEAD	45.5	36.5	38.5± 0.5	17.9	14.9	15.4±0.5	27.8	29.2	26.8±0.5
SVM	19.0	48.8	24.7±0.8	7.5	21.1	10.0±0.5	32.7	34.3	31.4±0.4
LogReg	26.9	34.5	28.7±0.6	6.4	9.9	7.3±0.4	12.2	14.9	12.7±0.5
LogReg^r	28.0	34.8	29.4±0.6	6.9	10.4	7.8±0.4	12.1	14.5	12.5±0.5
HAN	31.0	42.8	33.7±0.7	11.2	17.8	12.7±0.5	26.9	34.1	32.4±0.5
HAN+pretrainT	32.2	42.4	34.4±0.7	11.5	17.5	12.9±0.5	29.6	35.8	32.2±0.5
HAN+pretrainU	32.1	42.1	33.8±0.7	11.6	17.6	12.9±0.5	30.1	35.6	32.3±0.5
HAN^r	38.1	40.5	**37.8**±0.5	14.0	17.1	**14.7**±0.5	32.5	34.4	**33.4**±0.5
HAN+pretrainT^r	37.9	40.4	**37.6**±0.5	13.5	16.8	**14.4**±0.5	32.5	34.4	**33.4**±0.5
HAN+pretrainU^r	37.9	40.4	**37.6**±0.5	13.6	16.9	**14.4**±0.5	33.9	33.8	**33.8**±0.5

Table 1: Results of thread summarization. 'HAN' models are our proposed approaches adapted from the hierarchical attention networks (Yang et al. 2016). The models can be pretrained using unlabeled threads from TripAdvisor ('T') and Ubuntuforum ('U'). r indicates a redundancy removal step is applied. We report the variance of F-scores across all threads ('±'). A redundancy removal step improves recall scores (shown in gray) of the HAN models and boosts performance.

Carenini, Ng, and Zhou 2008; Murray and Carenini 2008; Oya and Carenini 2014). Many of these are driven by the publicly available Enron email corpus (Klimt and Yang 2004) and other mailing lists. Supervised approaches to email summarization draw on features such as sentence length, position, subject, sender/receiver, etc. Maximum entropy, SVM, CRF and variants (Ding et al. 2008) are used as classifiers. Further, Uthus and Aha (2011) described the opportunities and challenges of summarizing military chats. Giannakopoulos et al. (2015) presented a shared task on summarizing the comments found on news providers. We expect the human summaries created in this work will enable development of new approaches for thread summarization.

A recent strand of research is to model abstractive summarization (e.g., headline generation) as a sequence to sequence learning task (Rush, Chopra, and Weston 2015; Wiseman and Rush 2016; Nallapati et al. 2016). The models use an encoder to read a large chunk of input text and a decoder to generate a sentence one word at a time. Training the models require a large data collection where headlines are paired up with the first sentence of the articles. In contrast, our approach focuses on developing effective sentence and thread encoders and require less training data.

Conclusion

Supervised summarization approaches provide a promising avenue for scoring sentences. We have developed a class of supervised models by adapting the hierarchical attention networks to forum thread summarization. We compare the model with a range of unsupervised and supervised summarization baselines. Our experimental results demonstrate that the model performs better than most baselines and has the ability to capture contextual information with the recurrent structure. In particular, we believe that the incorporation of a redundancy removal step to supervised models is the key contributor to the results.

References

Anderson, A.; Huttenlocher, D.; Kleinberg, J.; and Leskovec, J. 2012. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).*

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Berg-Kirkpatrick, T.; Gillick, D.; and Klein, D. 2011. Jointly learning to extract and compress. In *Proceedings of ACL.*

Bhatia, S.; Biyani, P.; and Mitra, P. 2014. Summarizing online forum discussions – Can dialog acts of individual messages help? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Bhatia, S.; Biyani, P.; and Mitra, P. 2016. Identifying the role of individual user messages in an online discussion and its applications in thread retrieval. *Journal of the Association for Information Science and Technology (JASIST)* 67(2):276–288.

Boudin, F.; Mougard, H.; and Favre, B. 2015. Concept-based summarization using integer linear programming: From concept pruning to multiple optimal solutions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Cao, Z.; Li, W.; Li, S.; and Wei, F. 2017. Improving multi-document summarization via text classification. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI).*

Carenini, G.; Ng, R. T.; and Zhou, X. 2008. Summarizing emails with conversational cohesion and subjectivity. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chen, D.; Bolton, J.; and Manning, C. D. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of ACL*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of NIPS 2014 Workshop on Deep Learning*.

Dang, H. T., and Owczarzak, K. 2008. Overview of the TAC 2008 update summarization task. In *Proceedings of Text Analysis Conference (TAC)*.

Ding, Y., and Jiang, J. 2015. Towards opinion summarization from online forums. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*.

Ding, S.; Cong, G.; Lin, C.-Y.; and Zhu, X. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. In *Proceedings of ACL*.

Erkan, G., and Radev, D. R. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9:1871–1874.

Giannakopoulos, G.; Kubina, J.; Conroy, J. M.; Steinberger, J.; Favre, B.; Kabadjov, M.; Kruschwitz, U.; and Poesio, M. 2015. MultiLing 2015: Multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In *Proceedings of SIGDIAL*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Klimt, B., and Yang, Y. 2004. The enron corpus: A new dataset for email classification research. In *Proceedings of ECML*.

Li, J.; Luong, M.-T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lin, C.-Y. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of ACL Workshop on Text Summarization Branches Out*.

Luo, W.; Liu, F.; Liu, Z.; and Litman, D. 2016. Automatic summarization of student course feedback. In *Proceedings of NAACL*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Murray, G., and Carenini, G. 2008. Summarizing spoken and written conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Nallapati, R.; Zhou, B.; dos Santos, C.; Gulcehre, C.; and Xiang, B. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Oya, T., and Carenini, G. 2014. Extractive summarization and dialogue act modeling on email threads: An integrated probabilistic approach. In *Proceedings of the Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*.

Radev, D. R.; Jing, H.; Styś, M.; and Tam, D. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management* 40(6):919–938.

Rambow, O.; Shrestha, L.; Chen, J.; and Lauridsen, C. 2004. Summarizing email threads. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Ren, Z.; Ma, J.; Wang, S.; and Liu, Y. 2011. Summarizing web forum threads based on a latent topic propagation process. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Stephen, A. T., and Galak, J. 2012. The effects of traditional and social earned media on sales: A study of a microlending marketplace. *Journal of Marketing Research* 49.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.

Uthus, D. C., and Aha, D. W. 2011. Plans toward automated chat summarization. In *Proceedings of the ACL Workshop on Automatic Summarization for Different Genres, Media, and Languages*.

Vanderwende, L.; Suzuki, H.; Brockett, C.; and Nenkova, A. 2007. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Management* 43(6):1606–1618.

Wan, S., and McKeown, K. 2004. Generating overview summaries of ongoing email thread discussions. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*.

Wiseman, S., and Rush, A. M. 2016. Sequence-to-sequence learning as beam-search opimization. In *Proceedings of Empirical Methods on Natural Language Processing (EMNLP)*.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Yin, W.; Ebert, S.; and Schutze, H. 2016. Attention-based convolutional neural network for machine comprehension. In *Proceedings of the NAACL Workshop on Human-Computer Question Answering*.