This Email Could Save Your Life: Introducing the Task of Email Subject Line Generation

Rui Zhang *
Yale University
r.zhang@yale.edu

Joel Tetreault
Grammarly
joel.tetreault@grammarly.com

Abstract

Given the overwhelming number of emails, an effective subject line becomes essential to better inform the recipient of the email's content. In this paper, we propose and study the task of *email subject line generation*: automatically generating an email subject line from the email body. We create the first dataset for this task and find that email subject line generation favor extremely abstractive summary which differentiates it from news headline generation or news single document summarization. We then develop a novel deep learning method and compare it to several baselines as well as recent state-of-the-art text summarization systems. We also investigate the efficacy of several automatic metrics based on correlations with human judgments and propose a new automatic evaluation metric. Our system outperforms competitive baselines given both automatic and human evaluations. To our knowledge, this is the first work to tackle the problem of effective email subject line generation.

1 Introduction

Email is a ubiquitous form of online communication. An email message consists of two basic elements: an *email subject line* and an *email body*. The subject line, which is displayed to the recipient in the list of inbox messages, should tell what the email body is about and what the sender wants to convey. An effective email subject line becomes essential since it can help people manage a large number of emails. Table 1 shows an email body with three possible subject lines.

There have been several research tracks around email usage. While much effort has been focused on email summarization (Muresan et al., 2001; Nenkova and Bagga, 2003; Rambow et al., 2004), email keyword extraction and action detection (Turney, 2000; Lahiri et al., 2017; Lin et al., 2018), and email classification (Prabhakaran and Rambow, 2014; Alkhereyf and Rambow, 2017), to our knowledge there is no previous work on generating email subjects. In this paper, we propose the task of Subject Line Generation (SLG): automatically producing email subjects given the email body. While this is similar to email summarization, the two tasks serve different purposes in the process of email composition and consumption. A subject line is required when the sender writes the email, while a summary is more useful for long emails to benefit the recipient. An automatically generated email subject can also be used for downstream applications such as email triaging to help people manage emails more efficiently. Furthermore, while being similar to news headline generation or news single document summarization, email subjects are generally much shorter, which means a system must have the ability to summarize with a high compression ratio (Table 2). Therefore, we believe this task can also benefit other highly abstractive summarization such as generating section titles for long documents to improve reading comprehension speed and accuracy.

To introduce the task, we build the first dataset, Annotated Enron Subject Line Corpus (AESLC), by leveraging the Enron Corpus (Klimt and Yang, 2004) and crowdsourcing. Furthermore, in order

Email Body: Hi All, I would be grateful if you could get to me today via email a job description for your current role. I would like to get this to the immigration attorneys so that they can finalise the paperwork in preparation for INS filing once the UBS deal is signed. Kind regards, **Subject 1:** Current Job Description Needed *(COMMENT: This is good because it is both informative and succinct.)* **Subject 2:** Job Description *(COMMENT: This is okay but not informative enough.)* **Subject 3:** Request *(COMMENT: This is bad because it does not contain any specific information about the request.)*

Table 1: An email with three possible subject lines.

* Work done during the internship at Grammarly.

Dataset	domain	docs (train/val/test)	avg doc words	avg summary words
CNN (Cheng and Lapata, 2016)	News	90,266/1,220/1,093	760	46
XSum (Narayan et al., 2018a)	News	204,045/11,332/11,334	431	23
Gigaword News Headline (Rush et al., 2015)	News	3,799,588/394,622/381,197	31	8
Annotated Enron Subject Line Corpus	Business/Personal	14,436/1,960/1,906	75	4

Table 2: Annotated Enron Subject Line Corpus compared with other datasets.

to properly evaluate the subject, we use a combination of automatic metrics from the text summarization and machine translation fields, in addition to building our own regression-based Email Subject Quality Estimator (ESQE). Third, to generate effective email subjects, we propose a method which combines extractive and abstractive summarization using a two-stage process by *Multi-Sentence Selection and Rewriting with Email Subject Quality Estimation Reward*. The multi-sentence extractor first selects multiple sentences from the input email body. Extracted sentences capture salient information for writing a subject such as named entities and dates. Thereafter, the multi-sentence abstractor rewrites multiple selected sentences into a succinct subject line while preserving key information. For training the network, we use a multi-stage training strategy incorporating both supervised cross-entropy training and reinforcement learning (RL) by optimizing the reward provided by the ESQE model.

Our contributions are threefold: (1) We introduce the task of email subject line generation (SLG) and build a benchmark dataset AESLC.[1] (2) We investigate possible automatic metrics for SLG and study their correlations with human judgments. We also introduce a new email subject quality estimation metric (ESQE). (3) We propose a novel model to generate email subjects. Our automatic and human evaluations demonstrate that our model outperforms competitive baselines and approaches human-level quality.

2 Annotated Enron Subject Line Corpus

To prepare our email subject line dataset, we use the Enron dataset (Klimt and Yang, 2004) which is a collection of email messages of employees in the Enron Corporation. We use Enron because it can be released to the public and it contains business and personal type emails for which the subject line is already well-defined and useful. As shown in Table 2, email subjects are typically much shorter than summaries generated in previous news datasets. While being similar to news headline generation (Rush et al., 2015), email subject generation is also more challenging in the sense that it deals with different types of email subjects while the first sentence of a news article is often already a good headline and summary.

2.1 Data Preprocessing

The original Enron dataset contains 517,401 email messages from 150 user mailboxes. To extract body and subject pairs from the dataset, we take all messages from the inbox and sent folders of all mailboxes. We then perform email body cleaning, email filtering, and email de-duplication.

We first remove any content from the email body that has not been written by the author of the email. This includes automatically appended boilerplate material such as advertisements, attachments, legal disclaimers etc. Since we are interested in emails with enough information to generate meaningful subjects, we only keep emails with at least 3 sentences and 25 words in the email body. Furthermore, to ensure that the email subject truly corresponds to the content in the email body, we only take the first email of a thread and exclude replies or forward emails. So we filter out follow up messages which contain "Original Message" section in the email body or have subject lines starting with "RE:" (reply-to messages) or "FW:" (forward messages). Finally, we observe that the same message can be sent to multiple recipients so we remove duplicate emails to make sure there is no overlap between the train and test set. We only keep the subject and body while other information such as the sender/recipient identity can be incorporated in future work.

2.2 Subject Annotation

We noted that using only the original subject lines as references may be problematic for automatic evaluation purposes. First, there can be many different valid, effective subject lines for the same email, yet the original email subject is only one of them. This is similar to why automatic machine translation evaluation often relies on mul-

[1] dataset available at https://github.com/ryanzhumich/AESLC

tiple references. Second, the email subject may be too general or too vague when the sender does not put that much effort into writing. Third, the sender may assume some shared knowledge with the recipient so that the email subject contains information that cannot be found in the email body.

To address the issues above, we ask workers on Amazon Mechanical Turk to read Enron emails in our dev and test sets and write an appropriate subject line. Each email is annotated with 3 subject lines from 3 different annotators. For quality control, we manually review and reject improper email subjects such as empty subject lines, subject lines with typos, and subject lines that are too general or too vague, e.g., "Update", "Schedule", "Attention to Detail" because they contain no body-specific information and can be applied generically to many emails. We found that while three annotations are different, they often contain common keywords. To further quantify the variation among human annotations, we compute ROUGE-L F1 scores for each pair of annotations: 34.04, 33.38, 34.26.

3 Our Model

Our model is illustrated in Figure 1. Based on recent progress in news summarization (Chen and Bansal, 2018), our model generates email subjects in two stages: (1) The extractor selects multiple sentences containing salient information for writing a subject (§3.1). (2) The abstractor rewrites multiple selected sentences into a succinct subject line while preserving key information (§3.2).

We employ a multi-stage training strategy (§3.4) including a Reinforcement Learning (RL) phase because of its usefulness for text generation tasks (Ranzato et al., 2016; Bahdanau et al., 2017) to optimize the non-differentiable metrics such as ROUGE and METEOR. However, unlike ROUGE for summarization or METEOR for machine translation, there is no available automatic metric designed for email subject generation. Motivated by recent work on regression-based metrics for machine translation (Shimanaka et al., 2018) and dialog response generation (Lowe et al., 2017), we build a neural network (ESQE) to estimate the quality of an email subject given the email body (§3.3). The estimator is pretrained and fixed during RL training phase to provide rewards for the extractor agent.

While our model is based on Chen and Bansal (2018), they assume that there is a one-to-one relationship between the summary sentence and the document sentence: every summary sentence can be rewritten from exactly one sentence in the document. They also use ROUGE to make extraction labels and to provide rewards in their RL training phase. In contrast, our model extracts multiple sentences and rewrites them together into a single subject line. We also use word overlap to make extraction labels and use our novel ESQE as a reward function.

3.1 Multi-sentence Extractor

For the first stage, we need to select multiple sentences from the email body which contain the necessary information for writing a subject. This task can be formulated as a sequence-to-sequence learning problem where the output sequence corresponds to the position of "positive" sentences in the input email body. Therefore, we use a pointer network (Vinyals et al., 2015) to first build hierarchical sentence representations during encoding and then extract "positive" sentences during decoding.

Suppose our input is an email body D which consists of $|D|$ sentences:

$$D = [d_1, d_2, \ldots, d_j, \ldots, d_{|D|}]$$

We first use a temporal CNN (Kim, 2014) to build individual sentence representations. For each sentence, we feed the sequence of its word vectors into 1-D convolutional filters with various window sizes. We then apply ReLU activation and then max-over-time pooling. The sentence representation is a concatenation of activations from all filters

$$\mathbf{c}_j = \text{CNN}(d_j), j = 1, \ldots, |D| \quad (1)$$

Then we use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to capture document-level inter-sentence information over CNN outputs:

$$\begin{aligned} \overrightarrow{\mathbf{d}}_j &= \text{LSTM}^{\text{forward}}(\overrightarrow{\mathbf{d}}_{j-1}, \mathbf{c}_j) \\ \overleftarrow{\mathbf{d}}_j &= \text{LSTM}^{\text{backward}}(\overleftarrow{\mathbf{d}}_{j+1}, \mathbf{c}_j) \\ \mathbf{d}_j &= [\overrightarrow{\mathbf{d}}_j, \overleftarrow{\mathbf{d}}_j] \end{aligned} \quad (2)$$

For sentence extraction, another LSTM as decoder outputs one "positive" sentence at each time step t. Denoting the decoder hidden state as $\mathbf{h}^t$,

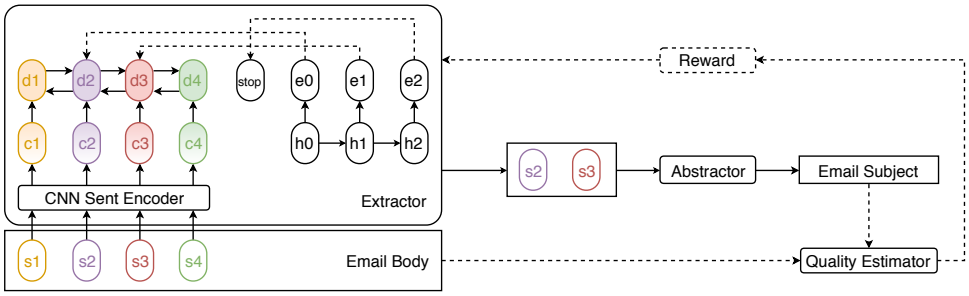

Figure 1: Our model architecture. In this example, the input email body consists of four sentences from which the extractor selects the second and the third. The abstractor generates an email subject from the selected sentences. The quality estimator provides rewards by scoring the subject against the email body.

we choose a "positive" sentence from a 2-hop attention process. First, we build a context vector $\mathbf{e}^t$ by attending all $\mathbf{d}_j$:

$$\begin{aligned} \hat{\alpha}_j^t &= \mathbf{v}_e^\intercal \tanh(\mathbf{W}_e \mathbf{d}_j + \mathbf{U}_e \mathbf{h}^t) \\ \alpha^t &= \text{softmax}(\hat{\alpha}^t) \\ \mathbf{e}^t &= \sum_j \alpha_j^t \mathbf{W}_e \mathbf{d}_j \end{aligned} \tag{3}$$

Then, we get an extraction probability distribution o^t over input sentences:

$$\begin{aligned} \hat{o}_j^t &= \mathbf{v}_o^\intercal \tanh(\mathbf{W}_o \mathbf{d}_j + \mathbf{U}_o \mathbf{e}^t) \\ P(o^t | o^1, o^2, \ldots, o^{t-1}) &= \text{softmax}(\hat{o}^t) \end{aligned} \tag{4}$$

where $\{\mathbf{v}, \mathbf{W}, \mathbf{U}\}$ are trainable parameters.

We also add a trainable "stop" vector with the same dimension as the sentence representation. The decoder can choose to stop by pointing to this "stop" sentence.

3.2 Multi-sentence Abstractor

In the second stage, the abstractor takes the selected sentences from the extractor and rewrites them into an email subject. We implement the abstractor as a sequence-to-sequence encoder-decoder model with the bilinear multiplicative attention (Luong et al., 2015) and copy mechanism (See et al., 2017). The copy mechanism enables the decoder to copy words directly from the input document, which is helpful to generate accurate information verbatim even for out-of-vocabulary words.

3.3 Email Subject Quality Estimator

Since there is no established automatic metric for SLG, we build our own Email Subject Quality Estimator (ESQE). Given an email body D and a potential subject for the subject s, our quality estimator outputs a real-valued Subject Quality score $\text{SQ}(D, s)$. The email subject and the email body are fed to a temporal CNN.

$$\mathbf{D} = \text{CNN}(D), \mathbf{s} = \text{CNN}(s) \tag{5}$$

We concatenate the output of CNNs as the email body and subject pair representation. Then, a single layer feed-forward neural net follows to predict the quality score from the representation.

$$\text{SQ}(D, s) = \text{FFNN}([\mathbf{D}, \mathbf{s}]) \tag{6}$$

To train the estimator, we collect human evaluations on 3,490 email subjects. In order to expose the estimator to both good and bad examples, 2,278 of the 3,490 are the original subjects and the remaining 1,212 subjects are generated by an existing summarization system. Each subject has 3 human evaluation scores (the same human evaluation as explained in §4.1) and we train our estimator to regress the average.

The inter-annotator agreement is 0.64 by Pearson's r correlation. Even though there is no value range restriction for the estimator output, we found the scores returned by our ESQE after training are bounded from 0.0 to 4.0.

3.4 Multi-Stage Training

Supervised Pretraining. We pretrain the extractor and the abstractor separately using supervised learning. To this end, we first create "proxy" sentence labels by checking word overlap between the subject and the body sentence. For each sentence in the body, we label it as "positive" if there is some token overlap of non-stopwords with the subject, negative otherwise. The multi-sentence extractor is trained to predict "positive" sentences by minimizing the cross-entropy loss. For the multi-sentence abstractor, we create training examples by pairing the "positive" sentences and the

original subject in the training set. Then the abstractor is trained to generate the subject by maximizing the log-likelihood.

RL Training for Extractor. To formulate this RL task at this stage, we treat the extractor as an agent, while the abstractor is pretrained and fixed. The ESQE provides the reward by judging the output subject. At each time step t, it observes a state $s_t = (D, d_{o^{t-1}})$, and samples an action a_t to pick a sentence from the distribution in Equation 4:

$$a_t \sim \pi_\theta(s_t, a_t = j) = P(o^t = j) \tag{7}$$

where π_θ denotes the policy network described in Section 3.1 with a set of trainable parameters θ. The episode is finished in T actions until the extractor picks the "end-of-extraction" signal. Then, the abstractor generates a subject from the extracted sentences and the quality estimator calculates the score. The quality estimator is the reward received by the extractor:

$$r(a_{1:T}) = \text{SQ}(D, s) \tag{8}$$

For training, we maximize the expected reward:

$$\mathcal{L}(\theta) = \mathbb{E}_{a_{1:T} \sim \pi_\theta}[r(a_{1:T})] \tag{9}$$

with the following gradient given by the REINFORCE algorithm (Williams, 1992):

$$\begin{aligned} \nabla_\theta \mathcal{L}(\theta) &= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta (r - b)] \\ &\approx \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t)(r(a_{1:T}) - b_t) \end{aligned} \tag{10}$$

b_t is the baseline reward introduced to reduce the high variance of gradients. The baseline network has the same architecture as the decoder of the extractor. But it has another set of trainable parameters θ_b and predicts the reward by minimizing the following mean squared error:

$$\mathcal{L}(\theta_b) = (b_t - r)^2 \tag{11}$$

4 Experimental Setup

4.1 Evaluation

Automatic Evaluation. Since SLG is a new task, we analyze the usefulness of automatic metrics from sister tasks, and also use human evaluation. We first use automatic metrics from text summarization and machine translation: (1) ROUGE (Lin, 2004) including F1 scores of ROUGE-1, ROUGE-2, and ROUGE-L. (2) METEOR (Denkowski and Lavie, 2014). They all rely on one or more references and measure the similarity between the output and the reference. In addition, we include ESQE, which is a reference-less metric.

Human Evaluation. While those automatic scores are quick and inexpensive to calculate, only our quality estimator is designed for evaluation of subject line generation. Therefore, we also conduct an extensive human evaluation on the *overall* score and two aspects of email quality: informativeness and fluency. An email subject is *informative* if it contains accurate and consistent details with the body, and it is *fluent* if free of grammar errors. We show the email body along with different system outputs as potential subjects (the models are anonymous). For each subject and each aspect, the human judge chooses a rating from 1 for Poor, 2 for Fair, 3 for Good, 4 for Great. We randomly select 500 samples and have each rated by 3 human judges.

4.2 Baselines

To benchmark our method, we use several methods from the summarization field, including some recent state-of-the-art systems, because the email subject line can be viewed as a short summary of the email content. They can be clustered into two groups.

(1) Unsupervised extractive or/and abstractive summarization. LEAD-2 directly uses the first two sentences as the subject line. We choose lead-2 to include both the greeting and the first sentence of main content. **TextRank** (Mihalcea and Tarau, 2004) and **LexRank** (Erkan and Radev, 2004) are two graph-based ranking models to extract the most salient sentence as the subject line. **Shang et al. (2018)** use a graph-based framework to extract topics and then generate a single abstractive sentence for each topic under a budget constraint.

(2) Neural summarization using encoder-decoder networks with attention mechanisms. (Sutskever et al., 2014; Bahdanau et al., 2015). The Pointer-Generator Network from **See et al. (2017)** augments the standard encoder-decoder network by adding the ability to copy words from the source text and using the coverage loss to avoid repetitive generation. **Paulus et al. (2018)** propose neural intra-attention models with a mixed objec-

	Dev				Test			
	R-1	R-2	R-L	METEOR	R-1	R-2	R-L	METEOR
LEAD-2	11.28	4.61	10.48	10.76	11.00	4.33	10.20	11.27*
TextRank	11.12	3.75	10.15	9.19	11.32	3.88	10.14	10.64*
LexRank	13.02	4.96	11.89	10.84	12.46	4.62	11.37	11.56*
Shang et al. (2018)	10.56	3.28	9.92	6.17	10.40	3.09	9.77	6.15
See et al. (2017)	18.02	5.73	16.63	10.83	17.02	5.45	15.78	10.31
Paulus et al. (2018)	14.08	5.09	13.36	9.07	13.49	4.55	12.83	8.65
Hsu et al. (2018)	16.59	4.67	15.12	**13.22***	15.75	4.54	14.41	**12.49***
Narayan et al. (2018a)	13.52	3.27	13.33	4.64	12.60	3.09	12.52	4.66
Our System	**25.41**	**11.34**	**25.07**	9.83	**23.67**	**10.29**	**23.44**	9.37
Human Annotation	23.43*	9.71*	22.17	10.87*	23.90*	10.09*	22.75*	11.04*

(a) Against the original subject as reference.

	Dev				Test			
	R-1	R-2	R-L	METEOR	R-1	R-2	R-L	METEOR
LEAD-2	18.88	9.47	17.41	20.70	18.29	8.54	16.62	**20.23**
TextRank	18.29	8.04	16.45	17.00*	17.93	7.47	16.00	16.98*
LexRank	21.82	10.83*	19.78	**20.82**	20.84	9.57*	18.68	19.97
Shang et al. (2018)	16.28	6.14	15.07	12.12	16.11	5.50	14.88	11.81
See et al. (2017)	23.37	7.36	20.99	16.27*	23.31	7.28	20.83	15.68*
Paulus et al. (2018)	15.12	4.62	13.98	10.82	14.56	4.39	13.53	10.37
Hsu et al. (2018)	22.98	7.07	19.95	18.83	22.80	7.09	19.85	18.45
Narayan et al. (2018a)	11.33	1.45	11.14	4.90	11.53	1.37	11.40	5.04
Our System	**25.39**	**10.94**	**24.72**	13.04	**26.11**	**11.43**	**25.64**	13.52
Original Subject	24.38*	10.15*	23.00*	16.49*	24.57	10.40	23.15	14.08
Human Annotation	35.93	17.76	33.55	21.74	36.19	17.75	33.50	21.42

(b) Against two human annotations as reference.

Table 3: Automatic metric scores. **bold**: best. underlined: second best. * indicates there is no statistically significant difference from our system with $p < 0.01$ under a paired t-test.

tive of supervised training and policy learning. **Hsu et al. (2018)** extend the pointer-generator network by unifying the sentence-level attention and the word-level attention. **Narayan et al. (2018a)** use a topic-based convolutional neural network to generate extreme summarization for news documents. While they are quite successful in single document summarization, they are mostly extractive, exhibiting a small degree of abstraction (Narayan et al., 2018a). It is unclear how they perform to generate email subject lines of extremely abstractive summarization. We train these models on our dataset.

4.3 Implementation Details

Our Model. We pretrain 128-dimensional word2vec (Mikolov et al., 2013) on our corpus as initialization and update word embeddings during training. We use single layer bidirectional LSTMs with 256 hidden units in all models. The convolutional sentence encoders have filters with window sizes (3,4,5) and there are 100 filters for each size. The batch size is 16 for all training phases. We use the Adam optimizer (Kingma and Ba, 2015) with learning rates of 0.001 for supervised pretraining and 0.0001 for RL. We apply gradient clipping (Pascanu et al., 2013) with L2-norm of 2.0. The training is stopped early if the validation performance is not improved for 3 consecutive epochs. All experiments are performed on a Tesla K80 GPU. All submodels can converge within 1-2 hours and 10 epochs so the whole training takes about 4 hours.

Baselines. For **TextRank** and **LexRank**, we use the sumy[2] implementation which uses the snowball stemmer, the sentence and word tokenizer from NLTK[3]. For **Shang et al. (2018)**, we

[2] https://github.com/miso-belica/sumy
[3] https://www.nltk.org/

use their extension of the Multi-Sentence Compression Graph (MSCG) of Filippova (2010) and a budget of 10 words in the submodular maximization. We choose the number of communities from [1,2,3,4,5] based on the dev set and we find that 1 works best. For the Pointer-Generator Network from **See et al. (2017)**, we follow their implementation[4] and use a batch size 16. For **Paulus et al. (2018)**, we use an implementation from Keneshloo et al. (2018)[5]. We did not include the intra-temporal attention and the intra-decoder attention because they hurt the performance. For **Hsu et al. (2018)**, we follow their code[6] with a batch size 16. All training is early stopped based on the dev set performance.

5 Results and Discussion

5.1 Automatic Metric Evaluation

We report the automatic metric scores against the original subject and the subjects generated by Turkers (human annotations) as references in Tables 3a and 3b respectively. Table 4 also shows the ESQE scores. Overall, our method outperforms the other baselines in all metrics except METEOR. Other systems can achieve higher METEOR scores because METEOR emphasizes recall (recall weighted 9 times more than precision) and other extractive systems such as LexRank can generate longer sentences as subject lines.

In Table 3a, where the original subject is the singular reference, the score of our system is rated close to and even higher than the human annotation on both sets. This is because our system is trained on the original subject and is likely a better domain fit. In Table 3b, all systems use two human annotations as the reference to have a fair comparison to the human-to-human agreement in the last row. Our system output is actually rated a bit higher than the original subject. This is because the original subject can differ from the human annotation when the sender and the recipient share some background knowledge hidden from the email content. Furthermore, in the last row, the human-to-human agreement is much higher than all the system outputs and the original subject. This indicates that different annotators write subjects with a similar choice of words. In Table 4, ESQE still considers our system better than other baselines, while the human annotation has the best quality score.

Evaluation of sub-components. Our extractor captures salient information by selecting multiple sentences from the email body. We measure its performance as a classification problem against the "proxy" sentence labels as explained in Section 3.4. The overall precision and recall on the test set is 74% and 42%, respectively. Out of 1906 test examples, 691 examples have more than one sentence selected, and 1626 first sentences and 973 non-first sentences are extracted. Furthermore, during RL training phase, the dev ESQE score increases from 2.30 to 2.40.

[4] https://github.com/abisee/pointer-generator

[5] https://github.com/yaserkl/RLSeq2Seq

[6] https://github.com/HsuWanTing/unified-summarization

	Dev	Test
LEAD-2	1.56	1.55
TextRank (Mihalcea and Tarau, 2004)	1.59	1.59
LexRank (Erkan and Radev, 2004)	1.57	1.56
Shang et al. (2018)	2.10	2.09
See et al. (2017)	2.22	2.19
Paulus et al. (2018)	2.30	2.30
Hsu et al. (2018)	1.44	1.46
Narayan et al. (2018a)	1.53	1.54
Our System	**2.40**	**2.39**
Original Subject	2.52	2.51
Human Annotation	2.53	2.54

Table 4: ESQE score. Compared with our system, all other are statistically significant with $p < 0.01$ under a paired t-test.

	Overall	Informative	Fluent
Random	1.10*	1.45	2.21
See et al. (2017)	1.45*	1.98	1.61
Our System	**2.28**	**2.38**	**2.89**
Original Subject	2.56	2.66	3.11
Human Annotation	2.74*	3.07	2.94

Table 5: Human evaluation. * indicates the difference from our system is statistically significant with $p < 0.01$ under a paired t-test.

	Pearson's r	Spearman's ρ
ESQE	**0.49**	**0.46**
ROUGE-1 F1	0.44	0.43
METEOR	0.40	0.40
Inter-Rater Agreement	0.64	0.58

Table 6: Correlation analysis between the automatic scores and the human evaluation.

Email Body: Dear Rick, Thanks for speaking with me today. Here is the position description for the KWI President of the Americas Opportunity. I feel that this is a tremendous opportunity to be an integral player with a very exciting relatively early stage Applications Software company, in the very exciting and hot Energy Commodities Sector; They are already profitable, pre-IPO. This position has a great compensation package. Please get back to me if you have an interest or if you know someone who might be intrigued by this opportunity. Thanks, Dal Coger
Original Subject: KWI President of the Americas
Human Annotation: KWI President of the Americas Opportunity
See et al., ACL 2017: Position Description - the Americas Sector Opportunity
Our System: KWI President of the Americas Position

(a) **Email ID:** buy-r_inbox_321

Email Body: Attached for your information are the following two filings made at FERC on Monday on behalf of WPTF: 1.. Motion to Intervene and Protest of the Western Power Trading Forum. This was filed in connection witht the ISO status report filing dealing with creditworthiness issues. 2.. Motion to Intervene and Comments of the Western Power Trading Forum. This was filed in connection with the Reliant and Mirant filing of a joint Section 206 complaint on October 18, 2001. My thanks to those who responded to the drafts with comments and suggestions. Dan
Original Subject: Monday's FERC Filings
Human Annotation: Two Filings Made at FERC
See et al., ACL 2017: FERC filings - FERC power and at monday was filing
Our System: Western Power Trading Filings

(b) **Email ID:** dasovich-j_inbox_1473

Email Body: Hi Evening MBA students, If you plan to graduate this semester for a December 2001 degree, will you please come by the Evening MBA office soon (by Tuesday, September 25 at the latest) and fill out an Application for Candidacy form? We have your fall transcript to assist you in filling out the form. Since we need your original signature, an office visit is best. Thanks, congratulations, and see you!
Original Subject: Planning to graduate this semester?
Human Annotation: December 2001 degree
See et al., ACL 2017: December application(graduate) - September 25
Our System: December 2001 degree application

(c) **Email ID:** dasovich-j_inbox_123

Email Body: As our last day is Friday, November 30th, we would love to toast the good times and special memories that we have shared with you over the past five years. Please join us at Teala's (W. Dallas) on Thursday, November 29th, beginning at 5pm. Looking forward to being with you, Lara and Janel Lara Leibman
Original Subject: Farewell Drinks
Human Annotation: Our last day
See et al., ACL 2017: Friday 30th and day, W. Dallas - November
Our System: Teala's

(d) **Email ID:** arnold-j_inbox_153

Table 7: Case study. The sentences extracted by our model are underlined. (a)(b)(c): Our model can generate effective subjects by extracting and rewriting multiple sentences containing salient information. (d): Our model fails to generate reasonable subjects for the novel topic of "farewell" which is not seen during training.

5.2 Human Evaluation

Table 5 shows that our system is rated higher than the baselines on overall, informative, and fluent aspects. For overall scores, the baselines are all between 1.5 and 2.0, indicating the subjects are usually considered as poor or fair (recall that the scale is 1-4, with 4 being the highest). Our system is 2.28, while the original subject and human annotation are between 2.5 and 3.0. This means more than half of our system outputs are at least fair, and the original subject and human annotation are often good or great. We also find that in 89 out of 500 emails, our system outputs have ratings higher than or equal to the original and human annotated subjects. Furthermore, the raters prefer the human annotated subject to the original subject.

5.3 Metric Correlation Analysis

It is important to check if the automatic metric scores can truly reflect the generation quality and serve as valid metrics for subject line generation. Therefore, in Table 6, we investigate their correlations with the human evaluation. To this end, we take the average of three human ratings and then calculate Pearson's r and Spearman's ρ between

different automatic scores and the average human rating. We also report the inter-rater agreement in the last row by checking the correlation between the third human rating and the average of the other two. We find that the inter-rater agreement is moderate with 0.64 for Pearson's r and 0.58 for Spearman's ρ. We would recommend ESQE because it has the highest correlations while being reference-less.

5.4 Case Study

Table 7 shows examples of our model outputs. Our model works well by first picking multiple sentences containing information such as named entities and dates and then rewriting them into a succinct subject line preserving the key information. In Example 7a, our model extracts sentences with the name of the company and position "KWI President of the Americas". It also captures the importance of the opportunity for this position. Similarly, in Example 7b, our model identifies "Western Power Trading" for "filings". In Example 7c, our model identifies the date of degree "December 2011" and action item "application". However, we also found our model can fail on emails about novel topics, as in Example 7d where the topic is scheduling farewell drinks. Our model only captures the name of the restaurant but not the purpose and topic since it has not seen this kind of email in training.

6 Related Work

Past NLP email research has focused on summarization (Muresan et al., 2001; Nenkova and Bagga, 2003; Rambow et al., 2004; Corston-Oliver et al., 2004; Wan and McKeown, 2004; Carenini et al., 2007; Zajic et al., 2008; Carenini et al., 2008; Ulrich et al., 2009), keyword extraction and action detection (Turney, 2000; Bennett and Carbonell, 2005; Dredze et al., 2008; Scerri et al., 2010; Loza et al., 2014; Lahiri et al., 2017; Lin et al., 2018), and classification (Prabhakaran et al., 2014; Prabhakaran and Rambow, 2014; Alkhereyf and Rambow, 2017). However, we could not find any previous work on email subject line generation. The very first study on email summarization is Muresan et al. (2001) who reduce the problem to extracting salient phrases. Later, Nenkova and Bagga (2003), Rambow et al. (2004), Wan and McKeown (2004) deal with the problem of email thread summarization by the sentence extraction approach.

Another related line of research is natural language generation. Our task is most similar to single document summarization because the email subject line can be viewed as a short summary of the email content. Therefore, we use different summarization models as baselines with techniques such as graph-based extraction and compression, sequence-to-sequence neural abstractive summarization with the hierarchical attention, copy, and coverage mechanisms. In addition, RL has become increasingly popular for text generation to optimize the non-differentiable metrics and to reduce the exposure bias in the traditional "teaching forcing" supervised training (Ranzato et al., 2016; Bahdanau et al., 2017; Zhang and Lapata, 2017; Sakaguchi et al., 2017). For example, Narayan et al. (2018b) use RL for ranking sentences in pure extractive summarization.

Furthermore, current methods on news headline generation (Lopyrev, 2015; Tilk and Alumäe, 2017; Kiyono et al., 2017; Tan et al., 2017; Shen et al., 2017) most follow the encoder-decoder model, while our model uses a multi-sentence selection and rewriting framework.

7 Conclusions and Future Work

In this paper, we introduce the task of email subject line generation. We build a benchmark dataset (AESLC) with crowdsourced human annotations on the Enron corpus and evaluate automatic metrics for this task. We propose our model of subject generation by Multi-Sentence Selection and Rewriting with Email Subject Quality Estimation Reward. Our model outperforms several competitive baselines and approaches human-level performance.

In the future, we would like to generalize it to multiple domains and datasets. We are also interested in generating more effective and appropriate subjects by incorporating prior email conversations, social context, the goal and style of emails, personality, among others.

Acknowledgements

We would like to thank Jimmy Nguyen and Vipul Raheja for their help in the data creation process. We also thank Dragomir Radev, Courtney Napoles, Dimitrios Alikaniotis, Claudia Leacock, Junchao Zheng, Maria Nadejde, Adam Faulkner, and three anonymous reviewers for their helpful discussion and feedback.

References

Sakhar Alkhereyf and Owen Rambow. 2017. Work hard, play hard: Email classification on the avocado and enron corpora. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Paul N Bennett and Jaime Carbonell. 2005. Detecting action-items in e-mail. In *SIGIR*.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *WWW*.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2008. Summarizing emails with conversational cohesion and subjectivity. *ACL*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *ACL*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *ACL*.

Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. *Text Summarization Branches Out*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*.

Mark Dredze, Hanna M Wallach, Danny Puller, and Fernando Pereira. 2008. Generating summary keywords for emails using topics. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 199–206. ACM.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *journal of artificial intelligence research*, 22:457–479.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *COLING*.

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

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *ACL*.

Yaser Keneshloo, Tian Shi, Chandan K Reddy, and Naren Ramakrishnan. 2018. Deep reinforcement learning for sequence to sequence models. *arXiv preprint arXiv:1805.09461*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Shun Kiyono, Sho Takase, Jun Suzuki, Naoaki Okazaki, Kentaro Inui, and Masaaki Nagata. 2017. Source-side prediction for neural headline generation. *arXiv preprint arXiv:1712.08302*.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *ECML*.

Shibamouli Lahiri, Rada Mihalcea, and P-H Lai. 2017. Keyword extraction from emails. *Natural Language Engineering*, 23(2).

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Chu-Cheng Lin, Dongyeop Kang, Michael Gamon, Madian Khabsa, Ahmed Hassan Awadallah, and Patrick Pantel. 2018. Actionable email intent modeling with reparametrized rnns. In *AAAI*.

Konstantin Lopyrev. 2015. Generating news headlines with recurrent neural networks. *arXiv preprint arXiv:1512.01712*.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *ACL*.

Vanessa Loza, Shibamouli Lahiri, Rada Mihalcea, and Po-Hsiang Lai. 2014. Building a dataset for summarization and keyword extraction from emails. In *LREC*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Smaranda Muresan, Evelyne Tzoukermann, and Judith L Klavans. 2001. Combining linguistic and machine learning techniques for email summarization. In *CoNLL*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *EMNLP*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. In *NAACL*.

Ani Nenkova and Amit Bagga. 2003. Facilitating email thread access by extractive summary generation. *RANLP*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *ICML*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *ICLR*.

Vinodkumar Prabhakaran and Owen Rambow. 2014. Predicting power relations between participants in written dialog from a single thread. In *ACL*.

Vinodkumar Prabhakaran, Emily E Reid, and Owen Rambow. 2014. Gender and power: How gender and gender environment affect manifestations of power. In *EMNLP*.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads. In *NAACL*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Grammatical error correction with neural reinforcement learning. In *IJCNLP*.

Simon Scerri, Gerhard Gossen, Brian Davis, and Siegfried Handschuh. 2010. Classifying action items for semantic email. In *LREC*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *ACL*.

Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, Mao-Song Sun, et al. 2017. Recent advances on neural headline generation. *Journal of Computer Science and Technology*.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. Metric for automatic machine translation evaluation based on universal sentence representations. In *NAACL: Student Research Workshop*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From neural sentence summarization to headline generation: a coarse-to-fine approach. In *IJCAI*.

Ottokar Tilk and Tanel Alumäe. 2017. Low-resource neural headline generation. *arXiv preprint arXiv:1707.09769*.

Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information retrieval*, 2(4):303–336.

Jan Ulrich, Giuseppe Carenini, Gabriel Murray, and Raymond T Ng. 2009. Regression-based summarization of email conversations. In *ICWSM*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *NIPS*.

Stephen Wan and Kathy McKeown. 2004. Generating overview summaries of ongoing email thread discussions. In *COLING*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

David M Zajic, Bonnie J Dorr, and Jimmy Lin. 2008. Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing & Management*, 44(4).

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *EMNLP*.