Neural Network-Based Abstract Generation for Opinions and Arguments

Lu Wang
College of Computer and Information Science
Northeastern University
Boston, MA 02115
luwang@ccs.neu.edu

Wang Ling
Google DeepMind
London, N1 0AE
lingwang@google.com

Abstract

We study the problem of generating abstractive summaries for opinionated text. We propose an attention-based neural network model that is able to absorb information from multiple text units to construct informative, concise, and fluent summaries. An importance-based sampling method is designed to allow the encoder to integrate information from an important subset of input. Automatic evaluation indicates that our system outperforms state-of-the-art abstractive and extractive summarization systems on two newly collected datasets of movie reviews and arguments. Our system summaries are also rated as more informative and grammatical in human evaluation.

1 Introduction

Collecting opinions from others is an integral part of our daily activities. Discovering what other people think can help us navigate through different aspects of life, ranging from making decisions on regular tasks to judging fundamental societal issues and forming personal ideology. To efficiently absorb the massive amount of opinionated information, there is a pressing need for automated systems that can generate concise and fluent opinion summary about an entity or a topic. In spite of substantial researches in opinion summarization, the most prominent approaches mainly rely on *extractive summarization* methods, where phrases or sentences from the original documents are selected for inclusion in the summary (Hu and Liu, 2004; Lerman et al., 2009). One of the problems that extractive methods suffer from is that they unavoidably include secondary or redundant information. On the contrary, *abstractive summarization* methods, which are able to generate text beyond the original input, can produce more coherent and concise summaries.

Movie: *The Martian*
Reviews:
- One the **smartest**, sweetest, and most satisfyingly suspenseful sci-fi films in years.
- ...an intimate sci-fi epic that is **smart**, spectacular and stirring.
- The Martian is a **thrilling**, human and moving sci-fi picture that is easily the most emotionally engaging film Ridley Scott has made...
- It's pretty sunny and often **funny**, a space oddity for a director not known for pictures with **a sense of humor**.
- The Martian **highlights the book's best qualities**, tones down its worst, and adds its own style...

Opinion Consensus (Summary): **Smart**, **thrilling**, and surprisingly **funny**, The Martian offers **a faithful adaptation of the bestselling book** that brings out the best in leading man Matt Damon and director Ridley Scott.

Topic: *This House supports the death penalty.*
Arguments:
- The state has a responsibility to protect the lives of innocent citizens, and enacting the death penalty may save lives by **reducing the rate of violent crime**.
- While the prospect of life in prison may be frightening, surely death is a more daunting prospect.
- A 1985 study by Stephen K. Layson at the University of North Carolina showed that a single execution **deters 18 murders**.
- Reducing the wait time on death row prior to execution can dramatically increase its **deterrent effect** in the United States.

Claim (Summary): The death penalty **deters crime**.

Figure 1: Examples for an opinion consensus of professional reviews (critics) about movie "*The Martian*" from www.rottentomatoes.com, and a claim about "death penalty" supported by arguments from idebate.org. Content with similar meaning is highlighted in the same color.

In this paper, we present *an attention-based neural network model for generating abstractive summaries of opinionated text*. Our system takes as input a set of text units containing opinions about the same topic (e.g. reviews for a movie, or arguments for a controversial social issue), and then outputs a one-sentence abstractive summary that describes the opinion consensus of the input.

Specifically, we investigate our abstract generation model on two types of opinionated text: *movie reviews* and *arguments on controversial topics*. Examples are displayed in Figure 1. The first example contains a set of professional reviews (or critics) about movie “The Martian” and an opinion consensus written by an editor. It would be more useful to automatically generate fluent opinion consensus rather than simply extracting features (e.g. plot, music, etc) and opinion phrases as done in previous summarization work (Zhuang et al., 2006; Li et al., 2010). The second example lists a set of arguments on “death penalty”, where each argument supports the central claim “death penalty deters crime”. Arguments, as a special type of opinionated text, contain reasons to persuade or inform people on certain issues. Given a set of arguments on the same topic, we aim at investigating the capability of our abstract generation system for the novel task of *claim generation*.

Existing abstract generation systems for opinionated text mostly take an approach that first identifies salient phrases, and then merges them into sentences (Bing et al., 2015; Ganesan et al., 2010). Those systems are not capable of generating new words, and the output summary may suffer from ungrammatical structure. Another line of work requires a large amount of human input to enforce summary quality. For example, Gerani et al. (2014) utilize a set of templates constructed by human, which are filled by extracted phrases to generate grammatical sentences that serve different discourse functions.

To address the challenges above, we propose to use an attention-based abstract generation model — a data-driven approach trained to generate informative, concise, and fluent opinion summaries. Our method is based on the recently proposed framework of neural encoder-decoder models (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014a), which translates a sentence in a source language into a target language. Different from previous work, our summarization system is designed to support multiple input text units. An attention-based model (Bahdanau et al., 2014) is deployed to allow the encoder to automatically search for salient information within context. Furthermore, we propose an importance-based sampling method so that the encoder can integrate information from an important subset of input text. The importance score of a text unit is estimated from a novel regression model with pairwise preference-based regularizer. With importance-based sampling, our model can be trained within manageable time, and is still able to learn from diversified input.

We demonstrate the effectiveness of our model on two newly collected datasets for movie reviews and arguments. Automatic evaluation by BLEU (Papineni et al., 2002) indicates that our system outperforms the state-of-the-art extract-based and abstract-based methods on both tasks. For example, we achieved a BLEU score of 24.88 on Rotten Tomatoes movie reviews, compared to 19.72 by an abstractive opinion summarization system from Ganesan et al. (2010). ROUGE evaluation (Lin and Hovy, 2003) also indicates that our system summaries have reasonable information coverage. Human judges further rated our summaries to be more informative and grammatical than compared systems.

2 Data Collection

We collected two datasets for movie reviews and arguments on controversial topics with gold-standard abstracts.[1] Rotten Tomatoes (`www.rottentomatoes.com`) is a movie review website that aggregates both professional critics and user-generated reviews (henceforth *RottenTomatoes*). For each movie, a one-sentence critic consensus is constructed by an editor to summarize the opinions in professional critics. We crawled 246,164 critics and their opinion consensus for 3,731 movies (i.e. around 66 reviews per movie on average). We select 2,458 movies for training, 536 movies for validation and 737 movies for testing. The opinion consensus is treated as the gold-standard summary.

We also collect an argumentation dataset from `idebate.org` (henceforth *Idebate*), which is a Wikipedia-style website for gathering pro and con arguments on controversial issues. The arguments under each debate (or topic) are organized into dif-

[1]The datasets can be downloaded from `http://www.ccs.neu.edu/home/luwang/`.

ferent "for" and "against" points. Each point contains a one-sentence central claim constructed by the editors to summarize the corresponding arguments, and is treated as the gold-standard. For instance, on a debate about "death penalty", one claim is "the death penalty deters crime" with an argument "enacting the death penalty may save lives by reducing the rate of violent crime" (Figure 1). We crawled 676 debates with 2,259 claims. We treat each sentence as an argument, which results in 17,359 arguments in total. 450 debates are used for training, 67 debates for validation, and 150 debates for testing.

3 The Neural Network-Based Abstract Generation Model

In this section, we first define our problem in Section 3.1, followed by model description. In general, we utilize a Long Short-Term Memory network for generating abstracts (Section 3.2) from a latent representation computed by an attention-based encoder (Section 3.3). The encoder is designed to search for relevant information from input to better inform the abstract generation process. We also discuss an importance-based sampling method to allow encoder to integrate information from an important subset of input (Sections 3.4 and 3.5). Post-processing (Section 3.6) is conducted to re-rank the generations and pick the best one as the final summary.

3.1 Problem Formulation

In summarization, the goal is to generate a summary y, composed by the sequence of words $y_1, ..., |y|$. Unlike previous neural encoder-decoder approaches which decode from only one input, our input consists of an arbitrary number of reviews or arguments (henceforth *text units* wherever there is no ambiguity), denoted as $x = \{x^1, ..., x^M\}$. Each text unit x^k is composed by a sequence of words $x_1^k, ..., x_{|x^k|}^k$. Each word takes the form of a representation vector, which is initialized randomly or by pre-trained embeddings (Mikolov et al., 2013), and updated during training. The summarization task is defined as finding $\hat{y}$, which is the most likely sequence of words $\hat{y}_1, ..., \hat{y}_N$ such that:

$$\hat{y} = argmax_y \log P(y|x) \tag{1}$$

where $\log P(y|x)$ denotes the conditional log-likelihood of the output sequence y, given the input text units x. In the next sections, we describe the attention model used to model $\log P(y|x)$.

3.2 Decoder

Similar as previous work (Sutskever et al., 2014b; Bahdanau et al., 2014), we decompose $\log P(y|x)$ into a sequence of word-level predictions:

$$\log P(y|x) = \sum_{j=1,...,|y|} \log P(y_j|y_1, ..., y_{j-1}, x) \tag{2}$$

where each word y_j is predicted conditional on the previously generated $y_1, ..., y_{j-1}$ and input x. The probability is estimated by standard word softmax:

$$p(y_j|y_1, ..., y_{j-1}, x) = softmax(\mathbf{h}_j) \tag{3}$$

$\mathbf{h}_j$ is the Recurrent Neural Networks (RNNs) state variable at timestamp j, which is modeled as:

$$\mathbf{h}_j = g(\mathbf{y}_{j-1}, \mathbf{h}_{j-1}, \mathbf{s}) \tag{4}$$

Here g is a recurrent update function for generating the new state $\mathbf{h}_j$ from the representation of previously generated word $\mathbf{y}_{j-1}$ (obtained from a word lookup table), the previous state $\mathbf{h}_{j-1}$, and the input text representation $\mathbf{s}$ (see Section 3.3).

In this work, we implement g using a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997), which has been shown to be effective at capturing long range dependencies. Here we summarize the update rules for LSTM cells, and refer readers to the original work (Hochreiter and Schmidhuber, 1997) for more details. Given an arbitrary input vector $\mathbf{u}_j$ at timestamp $j-1$ and the previous state $\mathbf{h}_{j-1}$, a typical LSTM defines the following update rules:

$$\begin{aligned}
\mathbf{i}_j &= \sigma(\mathbf{W}_{iu}\mathbf{u}_j + \mathbf{W}_{ih}\mathbf{h}_{j-1} + \mathbf{W}_{ic}\mathbf{c}_{j-1} + \mathbf{b}_i)\\
\mathbf{f}_j &= \sigma(\mathbf{W}_{fu}\mathbf{u}_j + \mathbf{W}_{fh}\mathbf{h}_{j-1} + \mathbf{W}_{fc}\mathbf{c}_{j-1} + \mathbf{b}_f)\\
\mathbf{c}_j &= \mathbf{f}_j \odot \mathbf{c}_{j-1} + \mathbf{i}_j \odot \tanh(\mathbf{W}_{cu}\mathbf{u}_j + \mathbf{W}_{ch}\mathbf{h}_{j-1} + \mathbf{b}_c)\\
\mathbf{o}_j &= \sigma(\mathbf{W}_{ou}\mathbf{u}_j + \mathbf{W}_{oh}\mathbf{h}_{j-1} + \mathbf{W}_{oc}\mathbf{c}_j + \mathbf{b}_o)\\
\mathbf{h}_j &= \mathbf{o}_j \odot \tanh(\mathbf{c}_j)
\end{aligned} \tag{5}$$

σ is component-wise logistic sigmoid function, and $\odot$ denotes Hadamard product. Projection matrices

$\mathbf{W}_{**}$ and biases $\mathbf{b}_*$ are parameters to be learned during training.

Long range dependencies are captured by the cell memory $\mathbf{c}_j$, which is updated linearly to avoid the vanishing gradient problem. It is accomplished by predicting two vectors $\mathbf{i}_j$ and $\mathbf{f}_j$, which determine what to keep and what to forget from the current timestamp. Vector $\mathbf{o}_j$ then decides on what information from the new cell memory $\mathbf{c}_j$ can be passed to the new state $\mathbf{h}_j$. Finally, the model concatenates the representation of previous output word $\mathbf{y}_{j-1}$ and the input representation $\mathbf{s}$ (see Section 3.3) as $\mathbf{u}_j$, which serves as the input at each timestamp.

3.3 Encoder

The representation of input text units $\mathbf{s}$ is computed using an attention model (Bahdanau et al., 2014). Given a single text unit $x_1, ..., x_{|x|}$ and the previous state $\mathbf{h_j}$, the model generates $\mathbf{s}$ as a weighted sum:

$$\sum_{i=1,...,|x|} a_i \mathbf{b}_i \tag{6}$$

where a_i is the attention coefficient obtained for word x_i, and $\mathbf{b}_i$ is the context dependent representation of x_i. In our work, we construct $\mathbf{b}_i$ by building a bidirectional LSTM over the whole input sequence $x_1, ..., x_{|x|}$ and then combining the forward and backward states. Formally, we use the LSTM formulation from Eq. 5 to generate the forward states $\mathbf{h}_1^f, ..., \mathbf{h}_{|x|}^f$ by setting $\mathbf{u}_j = \mathbf{x}_j$ (the projection word x_j using a word lookup table). Likewise, the backward states $\mathbf{h}_{|x|}^b, ..., \mathbf{h}_1^b$ are generated using a backward LSTM by feeding the input in the reverse order, that is, $\mathbf{u}_j = \mathbf{x}_{|x|-j+1}$. The coefficients a_i are computed with a softmax over all input:

$$a_i = softmax(v(\mathbf{b}_i, \mathbf{h}_{j-1})) \tag{7}$$

where function v computes the affinity of each word x_i and the current output context $\mathbf{h}_{j-1}$ — how likely the input word is to be used to generate the next word in summary. We set $v(\mathbf{b}_i, \mathbf{h}_{j-1}) = \mathbf{W}_s \cdot \tanh(\mathbf{W}_{cg}\mathbf{b}_i + \mathbf{W}_{hg}\mathbf{h}_{j-1})$, where $\mathbf{W}_*$ and $\mathbf{W}_{**}$ are parameters to be learned.

3.4 Attention Over Multiple Inputs

A key distinction between our model and existing sequence-to-sequence models (Sutskever et al., 2014b; Bahdanau et al., 2014) is that our input consists of multiple separate text units. Given an input of N text units, i.e. $\{x_1^k, ..., x_{|x^k|}^k\}_{k=1}^N$, a simple extension would be to concatenate them into one sequence as $z = x_1^1, ..., x_{|x^1|}^1, \text{SEG}, x_1^2, ..., x_{|x^2|}^2, \text{SEG}, x_1^N, ..., x_{|x^N|}^N$, where SEG is a special token that delimits inputs.

However, there are two problems with this approach. Firstly, the model is sensitive to the order of text units. Moreover, z may contain thousands of words. This will become a bottleneck for our model with a training time of $O(N|z|)$, since attention coefficients must be computed for all input words to generate each output word.

We address these two problems by sub-sampling from the input. The intuition is that even though the number of input text units is large, many of them are redundant or contain secondary information. As our task is to emphasize the main points made in the input, some of them can be removed without losing too much information. Therefore, we define an importance score $f(x^k) \in [0, 1]$ for each document x^k (see Section 3.5). During training, K candidates are sampled from a multinomial distribution which is constructed by normalizing $f(x^k)$ for input text units. Notice that the training process goes over the training set multiple times, and our model is still able to learn from more than K text units. For testing, top-K candidates with the highest importance scores are collapsed in descending order into z.

3.5 Importance Estimation

We now describe the importance estimation model, which outputs importance scores for text units. In general, we start with a ridge regression model, and add a regularizer to enforce the separation of summary-worthy text units from others.

Given a cluster of text units $\{x^1, ..., x^M\}$ and their summary y, we compute the number of overlapping content words between each text unit and summary y as its gold-standard importance score. The scores are uniformly normalized to $[0, 1]$. Each text unit x^k is represented as an $d-$dimensional feature vector $\mathbf{r_k} \in \mathbb{R}^d$, with label l_k. Text units in the training data are thus denoted with a feature matrix $\tilde{\mathbf{R}}$ and a label vector $\tilde{\mathbf{L}}$. We aim at learning $f(x^k) = \mathbf{r_k} \cdot \mathbf{w}$ by minimizing $||\tilde{\mathbf{R}}\mathbf{w} - \tilde{\mathbf{L}}||_2^2 + \beta \cdot ||\mathbf{w}||_2^2$. This is a standard formulation for ridge regression, and we use features in Table 1. Furthermore, pairwise preference constraints have been utilized for learning ranking models (Joachims, 2002). We then consider adding a *pairwise preference-based regularizing constraint* to incorporate a bias towards summary-worthy text units: $\lambda \cdot \sum_{\mathcal{T}} \sum_{x^p, x^q \in \mathcal{T}, l_p > 0, l_q = 0} ||(\mathbf{r_p} - \mathbf{r_q}) \cdot \mathbf{w} - 1||_2^2$, where $\mathcal{T}$ is a cluster of text units to be summarized. Term $(\mathbf{r_p} - \mathbf{r_q}) \cdot \mathbf{w}$ enforces the separation of summary-worthy text from the others. We further construct $\tilde{\mathbf{R}}'$ to contain all the pairwise differences $(\mathbf{r_p} - \mathbf{r_q})$. $\tilde{\mathbf{L}}'$ is a vector of the same size as $\tilde{\mathbf{R}}'$ with each element as 1. The objective function becomes:

$$J(\mathbf{w}) = ||\tilde{\mathbf{R}}\mathbf{w} - \tilde{\mathbf{L}}||_2^2 + \lambda \cdot ||\tilde{\mathbf{R}}'\mathbf{w} - \tilde{\mathbf{L}}'||_2^2 + \beta \cdot ||\mathbf{w}||_2^2 \quad (8)$$

λ, β are tuned on development set. With $\tilde{\boldsymbol{\beta}} = \beta \cdot \mathbf{I_d}$ and $\tilde{\boldsymbol{\lambda}} = \lambda \cdot \mathbf{I}_{|\mathbf{R}'|}$, **closed-form** solution for $\hat{\mathbf{w}}$ is:

$$\hat{\mathbf{w}} = (\tilde{\mathbf{R}}^{\mathbf{T}}\tilde{\mathbf{R}} + \tilde{\mathbf{R}}'^{\mathbf{T}}\tilde{\boldsymbol{\lambda}}\tilde{\mathbf{R}}' + \tilde{\boldsymbol{\beta}})^{-1}(\tilde{\mathbf{R}}^{\mathbf{T}}\tilde{\mathbf{L}} + \tilde{\mathbf{R}}'^{\mathbf{T}}\tilde{\boldsymbol{\lambda}}\tilde{\mathbf{L}}') \quad (9)$$

- num of words	- category in General Inquirer (Stone et al., 1966)
- unigram	- num of positive/negative/neutral words (General Inquirer, MPQA (Wilson et al., 2005))
- num of POS tags	
- num of named entities	
- centroidness (Radev, 2001)	
- avg/max TF-IDF scores	

Table 1: Features used for text unit importance estimation.

3.6 Post-processing

For testing phase, we re-rank the n-best summaries according to their cosine similarity with the input text units. The one with the highest similarity is included in the final summary. Uses of more sophisticated re-ranking methods (Charniak and Johnson, 2005; Konstas and Lapata, 2012) will be investigated in future work.

4 Experimental Setup

Data Pre-processing. We pre-process the datasets with Stanford CoreNLP (Manning et al., 2014) for tokenization and extracting POS tags and dependency relations. For RottenTomatoes dataset, we replace movie titles with a generic label in training, and substitute it with the movie name if there is any generic label generated in testing.

Pre-trained Embeddings and Features. The size of word representation is set to 300, both for input and output words. These can be initialized randomly or using pre-trained embeddings learned from Google news (Mikolov et al., 2013). We also extend our model with additional features described in Table 2. Discrete features, such as POS tags, are mapped into word representation via lookup tables. For continuous features (e.g TF-IDF scores), they are attached to word vectors as additional values.

- part of a named entity?	- category in General Inquirer
- capitalized?	- sentiment polarity (General Inquirer, MPQA)
- POS tag	
- dependency relation	- TF-IDF score

Table 2: Token-level features used for abstract generation.

Hyper-parameters and Stop Criterion. The LSTMs (Equation 5) for the decoder and encoders are defined with states and cells of 150 dimensions. The attention of each input word and state pair is computed by being projected into a vector of 100 dimensions (Equation 6).

Training is performed via Adagrad (Duchi et al., 2011). It terminates when performance does not improve on the development set. We use BLEU (up to 4-grams) (Papineni et al., 2002) as evaluation metric, which computes the precision of n-grams in generated summaries with gold-standard abstracts as the reference. Finally, the importance-based sampling rate (K) is set to 5 for experiments in Sections 5.2 and 5.3.

Decoding is performed by beam search with a beam size of 20, i.e. we keep 20 most probable output sequences in stack at each step. Outputs with `end of sentence` token are also considered for re-ranking. Decoding stops when every beam in stack generates the `end of sentence` token.

5 Results

5.1 Importance Estimation Evaluation

We first evaluate the importance estimation component described in Section 3.5. We compare with Support Vector Regression (SVR) (Smola and Vapnik, 1997) and two baselines: (1) a *length baseline* that ranks text units based on their length, and (2) a *centroid baseline* that ranks text units according to their centroidness, which is computed as the cosine similarity between a text unit and centroid of the cluster to be summarized (Erkan and Radev, 2004).

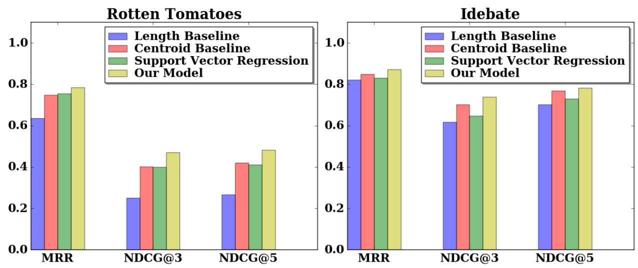

Figure 2: Evaluation of importance estimation by mean reciprocal rank (MRR), and normalized discounted cumulative gain at top 3 and 5 returned results (NDCG@3 and NDCG@5). Our regression model with pairwise preference-based regularizer uniformly outperforms baseline systems on both datasets.

We evaluate using mean reciprocal rank (MRR), and normalized discounted cumulative gain at top 3 and 5 returned results (NDCG@3). Text units are considered relevant if they have at least one overlapping content word with the gold-standard summary. From Figure 2, we can see that our importance estimation model produces uniformly better ranking performance on both datasets.

5.2 Automatic Summary Evaluation

For automatic summary evaluation, we consider three popular metrics. ROUGE (Lin and Hovy, 2003) is employed to evaluate n-grams recall of the summaries with gold-standard abstracts as reference. ROUGE-SU4 (measures unigram and skip-bigrams separated by up to four words) is reported. We also utilize BLEU, a precision-based metric, which has been used to evaluate various language generation systems (Chiang, 2005; Angeli et al., 2010; Karpathy and Fei-Fei, 2014). We further consider METEOR (Denkowski and Lavie, 2014). As a recall-oriented metric, it calculates similarity between generations and references by considering synonyms and paraphrases.

For comparisons, we first compare with an abstractive summarization method presented in Ganesan et al. (2010) on the RottenTomatoes dataset. Ganesan et al. (2010) utilize a graph-based algorithm to remove repetitive information, and merge opinionated expressions based on syntactic structures of product reviews.[2] For both datasets, we consider two extractive summarization approaches: (1) LEXRANK (Erkan and Radev, 2004) is an unsupervised method that computes text centrality based on PageRank algorithm; (2) Sipos et al. (2012) propose a supervised SUBMODULAR summarization model which is trained with Support Vector Machines. In addition, LONGEST sentence is picked up as a baseline.

Four variations of our system are tested. One uses randomly initialized word embeddings. The rest of them use pre-trained word embeddings, additional features in Table 2, and their combination. For all systems, we generate a one-sentence summary.

Results are displayed in Table 3. Our system with pre-trained word embeddings and additional features achieves the best BLEU scores on both datasets (in **boldface**) with statistical significance (two-tailed Wilcoxon signed rank test, $p < 0.05$). Notice that our system summaries are conciser (i.e. shorter on average), which lead to higher scores on precision based-metrics, e.g. BLEU, and lower scores on recall-based metrics, e.g. METEOR and ROUGE. On RottenTomatoes dataset, where summaries generated by different systems are similar in length, our system still outperforms other methods in METEOR and ROUGE in addition to their significantly better BLEU scores. This is not true on Idebate, since the length of summaries by extract-based systems is significantly longer. But the BLEU scores of our system are considerably higher. Among our four systems, models with pre-trained word embeddings in general achieve better scores. Though additional features do not always improve the performance, we find that they help our systems converge faster.

5.3 Human Evaluation on Summary Quality

For human evaluation, we consider three aspects: *informativeness* that indicates how much salient information is contained in the summary, *grammaticality* that measures whether a summary is grammatical, and *compactness* that denotes whether a summary contains unnecessary information. Each aspect is rated on a 1 to 5 scale (5 is the best). The judges are

[2] We do not run this model on Idebate because it relies on high redundancy to detect repetitive expressions, which is not observed on Idebate.

	RottenTomatoes				Idebate			
	Length	**BLEU**	**METEOR**	**ROUGE**	**Length**	**BLEU**	**METEOR**	**ROUGE**
Extract-Based Systems								
LONGEST	47.9	8.25	**8.43**	**6.43**	44.0	6.36	10.22	12.65
LEXRANK	16.7	19.93	5.59	3.98	26.5	13.39	9.33	10.58
SUBMODULAR	16.8	17.22	4.89	3.01	23.2	15.09	**10.76**	**13.67**
Abstract-Based Systems								
OPINOSIS	22.0	19.72	6.07	4.90	–	–	–	–
OUR SYSTEMS								
words	15.7	19.88	6.07	5.05	14.4	22.55*	7.38	8.37
words (pre-trained)	15.8	23.22*	*6.51*	*5.70*	13.9	23.93*	7.42	9.09
words + features	17.5	19.73	6.43	5.53	13.5	23.65*	7.33	7.79
words (pre-trained) + features	14.2	**24.88***	6.00	4.96	13.0	**25.84***	7.56	8.81

Table 3: Automatic evaluation results by BLEU, METEOR, and ROUGE SU-4 scores (multiplied by 100) for abstract generation systems. The average lengths for human written summaries are 11.5 and 24.6 for RottenTomatoes and Idebate. The best performing system for each column is highlighted in **boldface**, where our system with pre-trained word embeddings and additional features achieves the best BLEU scores on both datasets. Our systems that are statistically significantly better than the comparisons are highlighted with $*$ (two-tailed Wilcoxon signed rank test, $p < 0.05$). Our system also has the best METEOR and ROUGE scores (in *italics*) on RottenTomatoes dataset among learning-based systems.

	Info	**Gram**	**Comp**	**Avg Rank**	**Best%**
LEXRANK	3.4	4.5	4.3	2.7	11.5%
OPINOSIS	2.8	3.1	3.3	3.5	5.0%
OUR SYSTEM	**3.6**	**4.8**	4.2	**2.3**	**18.0%**
HUMAN ABSTRACT	4.2	4.8	4.5	1.5	65.5%

Table 4: Human evaluation results for abstract generation systems. Inter-rater agreement for overall ranking is 0.71 by Krippendorff's α. Informativeness (**Info**), grammaticality (**Gram**), and Compactness (**Comp**) are rated on a 1 to 5 scale, with 5 as the best. Our system achieves the best informativeness and grammaticality scores among the three learning-based systems. Our summaries are ranked as the best in 18% of the evaluations, and are also ranked higher than compared systems on average.

also asked to give a ranking on all summary variations according to their overall quality.

We randomly sampled 40 movies from RottenTomatoes test set, each of which was evaluated by 5 distinct human judges. We hired 10 proficient English speakers for evaluation. Three system summaries (LexRank, Opinosis, and our system) and human-written abstract along with 20 representative reviews were displayed for each movie. Reviews with the highest gold-standard importance scores were selected.

Results are reported in Table 4. As it can be seen, our system outperforms the abstract-based system OPINOSIS in all aspects, and also achieves better informativeness and grammaticality scores than LEXRANK, which extracts sentences in their original form. Our system summaries are ranked as the best in 18% of the evaluations, and has an average ranking of 2.3, which is higher than both OPINOSIS and LEXRANK on average. An inter-rater agreement of Krippendorff's α of 0.71 is achieved for overall ranking. This implies that our attention-based abstract generation model can produce summaries of better quality than existing summarization systems. We also find that our system summaries are constructed in a style closer to human abstracts than others. Sample summaries are displayed in Figure 3.

5.4 Sampling Effect

We further investigate whether taking inputs sampled from distributions estimated by importance scores trains models with better performance than the ones learned from fixed input or uniformly-sampled input. Recall that we sample K text units based on their importance scores (*Importance-Based Sampling*). Here we consider two other setups: one is sampling K text units uniformly from the input (*Uniform Sampling*), another is picking K text units with the highest scores (*Top K*). We try various K values. Results in Figure 4 demonstrates that Importance-Based Sampling can produce comparable BLEU scores to Top K methods, while both of them outperform Uniform Sampling. For METEOR score, Importance-Based Sampling uniformly outperforms the other two methods[3].

5.5 Further Discussion

Finally, we discuss some other observations and potential improvements. First, applying the re-ranking component after the model generates n-best abstracts leads to better performance. Preliminary experiments show that simply picking the top-1 gener-

[3]We observe similar results on the Idebate dataset

Movie: *The Neverending Story*
Reviews: (1) Here is a little adventure that fed on our uncultivated need to think, and wonder... (2) Magical storytelling targeted at children still fascinates. (3)...the art direction involved a lot of imagination.
Human: A magical journey about the power of a young boy's imagination to save a dying fantasy land, The Neverending Story remains a much-loved kids adventure.
LexRank: It pokes along at times and lapses occasionally into dark moments of preachy philosophy, but this is still a charming, amusing and harmless film for kids.
Opinosis: The Neverending Story is a silly fantasy movie that often shows its age .
Our System: The Neverending Story is an entertaining children's adventure, with heart and imagination to spare.

Movie: *Joe Strummer: The Future is Unwritten*
Reviews: (1) The late punk rock legend Joe Strummer is rendered fully human in Julian Temple's engrossing and all-encompassing portrait. (2) The movie fascinates not so much because of Strummer... but because of the way Temple organized and edited the film. (3) One of the most compelling documentary portraits of a musician yet made.
Human: Displaying Joe Strummer warts and all, The Future is Unwritten succeeds as both an engrossing documentary and a comprehensive examination of one of music's most legendary figures.
LexRank: Joe Strummer: The Future Is Unwritten is a film for fans – really big fans .
Opinosis: Joe Strummer: The Future Is Unwritten is for fans – really big fans .
Our System: Fascinating and insightful, Joe Strummer: The Future Is Unwritten is a thoroughly engrossing documentary.

Topic: *This House would detain terror suspects without trial.*
Arguments: (1) Governments must have powers to protect their citizens against threats to the life of the nation.(2) Everyone would recognise that rules that are applied in peacetime may not be appropriate during wartime.
Human: Governments must have powers to protect citizens from harm.
LexRank: This is not merely to directly protect citizens from political violence, but also because political violence handicaps the process of reconstruction in nation-building efforts.
Our System: Governments have the obligation to protect citizens from harmful substances.

Topic: *This House would replace Christmas with a festival for everyone.*
Arguments: (1) Christmas celebrations in the Western world... do not respect the rights of those who are not religious. (2) States should instead be sponsoring and celebrating events that everyone can join in equally, regardless of religion, race or class.
Human: States should respect the freedom from religion, as well as the freedom of religion.
LexRank: For school children who do not share the majority-Christian faith, Christmas celebrations require either their participation when they do not want to, through coercion, or their non-participation and therefore isolation whilst everyone else celebrations their inclusiveness.
Our System: People have a right to freedom of religion.

Figure 3: Sample summaries generated by different systems on movie reviews and arguments. We only show a subset of reviews and arguments due to limited space.

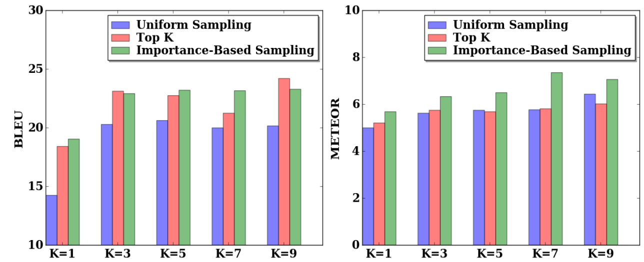

Figure 4: Sampling effect on RottenTomatoes.

ations produces inferior results than re-ranking them with simple heuristics. This suggests that the current models are oblivious to some task specific issues, such as informativeness. Post-processing is needed to make better use of the summary candidates. For example, future work can study other sophisticated re-ranking algorithms (Charniak and Johnson, 2005; Konstas and Lapata, 2012).

Furthermore, we also look at the difficult cases where our summaries are evaluated to have lower informativeness. They are often much shorter than the gold-standard human abstracts, thus the information coverage is limited. In other cases, some generations contain incorrect information on domain-dependent facts, e.g. named entities, numbers, etc. For instance, a summary "a poignant coming-of-age tale marked by a breakout lead performance from Cate Shortland" is generated for movie "Lore". This summary contains "Cate Shortland" which is the director of the movie instead of actor. It would require semantic features to handle this issue, which has yet to be attempted.

6 Related Work

Our work belongs to the area of opinion summarization. Constructing fluent natural language opinion summaries has mainly considered product reviews (Hu and Liu, 2004; Lerman et al., 2009), community question answering (Wang et al., 2014), and editorials (Paul et al., 2010). Extractive summarization approaches are employed to identify summary-worthy sentences. For example, Hu and Liu (2004) first identify the frequent product features and then attach extracted opinion sentences to the corresponding feature. Our model instead utilizes abstract generation techniques to construct natural language summaries. As far as we know, we are also the first to study claim generation for arguments.

Recently, there has been a growing interest in generating abstractive summaries for news articles (Bing et al., 2015), spoken meetings (Wang and Cardie, 2013), and product reviews (Ganesan et al., 2010; Di Fabbrizio et al., 2014; Gerani et al., 2014). Most approaches are based on phrase extraction, from which an algorithm concatenates them into sentences (Bing et al., 2015; Ganesan et al., 2010). Nevertheless, the output summaries are not guaranteed to be grammatical. Gerani et al. (2014) then design a set of manually-constructed realization templates for producing grammatical sentences that serve different discourse functions. Our approach does not require any human-annotated rules, and can be applied in various domains.

Our task is closely related to recent advances in neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014a). Based on the sequence-to-sequence paradigm, RNNs-based models have been investigated for compression (Filippova et al., 2015) and summarization (Filippova et al., 2015; Rush et al., 2015; Hermann et al., 2015) at sentence-level. Built on the attention-based translation model in Bahdanau et al. (2014), Rush et al. (2015) study the problem of constructing abstract for a single sentence. Our task differs from the models presented above in that our model carries out abstractive decoding from multiple sentences instead of a single sentence.

7 Conclusion

In this work, we presented a neural approach to generate abstractive summaries for opinionated text. We employed an attention-based method that finds salient information from different input text units to generate an informative and concise summary. To cope with the large number of input text, we deploy an importance-based sampling mechanism for model training. Experiments showed that our system obtained state-of-the-art results using both automatic evaluation and human evaluation.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. 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)*, pages 1587–1597, Beijing, China, July. Association for Computational Linguistics.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Giuseppe Di Fabbrizio, Amanda J Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. *INLG 2014*, page 54.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal, September. Association for Computational Linguistics.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on com-*

putational linguistics, pages 340–348. Association for Computational Linguistics.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613, Doha, Qatar, October. Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA. ACM.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, pages 1700–1709. ACL.

Andrej Karpathy and Li Fei-Fei. 2014. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*.

Ioannis Konstas and Mirella Lapata. 2012. Concept-to-text generation via discriminative reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 369–378, Jeju Island, Korea, July. Association for Computational Linguistics.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 514–522, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 653–661, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 71–78.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Michael J. Paul, ChengXiang Zhai, and Roxana Girju. 2010. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 66–76, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dragomir R. Radev. 2001. Experiments in single and multidocument summarization using mead. In *In First Document Understanding Conference*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September. Association for Computational Linguistics.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin learning of submodular summarization models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 224–233, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alex Smola and Vladimir Vapnik. 1997. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

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

In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014b. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, Sofia, Bulgaria, August. Association for Computational Linguistics.

Lu Wang, Hema Raghavan, Claire Cardie, and Vittorio Castelli. 2014. Query-focused opinion summarization for user-generated content. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1660–1669, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 43–50, New York, NY, USA. ACM.