Cross-domain Aspect/Sentiment-aware Abstractive Review Summarization*

Min Yang
SIAT, Chinese Academy of Sciences
min.yang@siat.ac.cn

Qiang Qu
SIAT, Chinese Academy of Sciences
qiang@siat.ac.cn

Jia Zhu
South China Normal University
jzhu@m.scnu.edu.cn

Ying Shen
Peking University Shenzhen Graduate School
shenying@pkusz.edu.cn

Zhou Zhao
Zhejiang University
zhaozhou@zju.edu.cn

ABSTRACT

This study takes the lead to study the aspect/sentiment-aware abstractive review summarization in domain adaptation scenario. The proposed model *CASAS* (neural attentive model for Cross-domain Aspect/Sentiment-aware Abstractive review Summarization) leverages domain classification task, working on datasets of both source and target domains, to recognize the domain information of texts and transfer knowledge from source domains to target domains. The extensive experiments on Amazon reviews demonstrate that *CASAS* outperforms the compared methods in both out-of-domain and in-domain setups.

KEYWORDS

Abstractive Review Summarization; Topic Modeling; Domain adaptation

ACM Reference Format:

Min Yang, Qiang Qu, Jia Zhu, Ying Shen, and Zhou Zhao. 2018. Cross-domain Aspect/Sentiment-aware Abstractive Review Summarization. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269273

1 INTRODUCTION

User generated reviews on products are expanding rapidly with the emergence of e-commerce. These reviews are valuable to business organizations for improving their products and to individual consumers for making informed decisions. Unfortunately, reading through all the product reviews is hard, especially for the reviews that are lengthy and ungrammatical. Therefore, automatically generating the coherent and concise summaries of user reviews is essential for the e-commerce platforms.

Despite the remarkable progress of neural abstractive text summarization [7, 8, 11], generating aspect/sentiment-aware summaries of product reviews remains a challenge. (i) Prior summarization models usually require a large number of labeled data. Nevertheless, annotating sufficient data is labor-intensive and time-consuming, establishing significant barriers for adapting the summarization systems to new domains which have limited labeled data. (ii) Neural sequence-to-sequence models tend to generate trivial and generic summary, often involving high-frequency phrases. These summaries cannot capture the aspect and sentiment information from the product reviews which play a vital role in helping customers to make quick and informed decisions on certain products [4].

To alleviate these limitations, we propose "*CASAS*", a neural attentive model for **C**ross-domain **A**spect/**S**entiment-aware **A**bstractive review **S**ummarization. *CASAS* is multi-task system, in which a document modeling module is shared across tasks. A domain classifier, the first subtask in *CASAS*, is trained for the reviews in both source and target domains. A weakly supervised LDA model (wsLDA) is proposed to learn domain-specific aspect and sentiment lexicon representations which are then used to calculate the aspect/sentiment-aware review representations via a soft attention mechanism. The domain classifier helps the semantic analysis and comprehension of documents which are expected to contribute to the knowledge transfer from source domains to target domains. The abstractive review summarizer, the second subtask in *CASAS*, shares the document modeling module with the domain classifier. The learned aspect/lexicon-aware review representations are fed into a pointer-generator network to generate aspect/sentiment-aware abstractive summaries of given reviews.

We summarize our main contributions as follows: (1) To our knowledge, this is the first work dealing with abstractive review summarization in **domain adaptation scenarios**. (2) We leverage text classification to learn better review representation and transfer knowledge from the source domain to the target domain. (3) *CASAS* integrates the supervised deep learning system with the unsupervised probabilistic generative model to strengthen the representation learning via an attention mechanism. The learned representation are expected to capture aspect and sentiment knowledge. (4) The experimental results show that our model outperforms the competitors from both quantitative and qualitative perspectives.

2 RELATED WORK

Inspired by the recent success of the encoder-decoder framework in statistical machine translation, there has been increasing interest in generalizing the neural language model to the field of abstractive summarization [7, 8, 10, 11]. For example, Rush et al. [10] were the

*The work was partially supported by CAS Pioneer Hundred Talents Program. Q. Qu is the corresponding author.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269273

first to apply attention-based encoder-decoder model to abstractive text summarization, achieving state-of-the-art performance on two sentence-level summarization datasets. See et al. [11] proposed a hybrid pointer-generator network that allowed both copying words from the source text and generating words from a fixed vocabulary.

Several recent studies attempted to integrate the encoder-decoder RNN and reinforcement learning paradigms for abstractive summarization, taking advantages of both [3, 8]. For example, Paulus et al. [8] combined the maximum-likelihood cross-entropy loss with rewards from policy gradient reinforcement learning to reduce exposure bias. Liu et al. [3] proposed an adversarial process for abstractive text summarization, in which the generator is built as an agent of reinforcement learning.

To date, no work is towards the aspect/sentiment-aware abstractive review summarization in **domain adaptation scenario**.

3 METHODOLOGY

We use X^s and X^t to denote the collection of reviews in source and target domains respectively. Each document in both source and target domains has a binary domain label y (e.g., Food or Electronics domains). Each document x^s in source domain has a corresponding reference summary sum^s.

Our model *CASAS* consists of two components: the domain classification model and the abstractive review summarization model, both working on a shared document encoding layer. In this section, we elaborate the main components of *CASAS* in detail.

3.1 Domain Classification in *CASAS*

In this subsection, we present a simple LSTM-based classification model for domain classification.

LSTM Encoder. We convert each document x=$\{w_1, \ldots, w_{N_x}\}$ into a sequence of hidden states H=$\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{N_x}\}$ by a single layer LSTM network, where N_x is the length of document x. The update of the hidden states of LSTM at time step t is computed as $h_t = \text{LSTM}(\boldsymbol{h}_{t-1}, w_t)$. The reader can refer to [2] for the details of LSTM.

Aspect and lexicon representation learning by wsLDA. In order to capture the relevant and discriminative information from a text piece in response to a given aspect and its sentiment polarity, we incorporate domain-specific aspect and sentiment lexicon background knowledge into the representation of a document. To that end, we first come up with a novel probabilistic generative model (wsLDA, weakly supervised LDA) to learn the domain-specific aspect and sentiment lexicon representations. Then we develop a soft-attention mechanism to combine aspect (sentiment lexicon) knowledge and document representations so that our model is able to attend the aspect information and the corresponding sentiment.

Because of its unsupervised nature, LDA [1] can be used as a generic tool to group words into categories. Inspired by [6, 12], our wsLDA model extends the standard LDA model, employing seed words to guide the topic model construction. The topics extract and categorize aspect terms and sentiment words automatically in the corresponding aspect and sentiment lexicon categories. It is thus able to best meet our specific needs, producing aspect and sentiment lexicon representations.

Suppose each document has three classes of topics: two *sentiment* topic, K *aspect* topics, and M *other* topics which are not related to sentiments and aspects, such as stopwords. Each topic is associated with a multinomial distribution over words. To prevent conceptual confusion, we use superscripts "*senti.*", "*aspect*" and "*other*" to indicate the variables that are related to *sentiment* topics, *aspect* topics and *other* topics, respectively. In addition, we assume that the vocabulary consists of V distinct words indexed by $\{1, \ldots, V\}$. For each document, we have three topic distributions $\theta^{senti.}$, θ^{aspect} and θ^{other} which represent the probabilities of *sentiment* topic *n*, *aspect* topic *k* and *other* topic *m*, respectively.

We use $\phi^{senti.}_{n,w}$, $\phi^{aspect}_{k,w}$ and $\phi^{other}_{m,w}$ to represent the probabilities of word w under *sentiment* topic n, *aspect* topic k and *other* topic m, respectively. The generation of *other* topics is similar to the original LDA model. For each *sentiment* topic n, its word distribution $\phi^{senti.}_n$ is chosen from a Dirichlet distribution $\text{Dir}(\beta^{senti.}_n)$, where $\beta^{senti.}_n$ is a V-dimensional hyperparameter that needs to be defined by users. The V-dimensional vector $\beta^{senti.}_n$ is computed by

$$\beta^{senti.}_{n,w} = \gamma_0(1-\omega_w) + \gamma_1\omega_w, \quad \text{for } w \in \{1, \ldots, V\} \tag{1}$$

where $\omega_w = 1$ if the word w is a seed word in *sentiment* topic n; otherwise, we set $\omega_w = 0$. The scalars γ_0 and γ_1 are hyperparameters. Intuitively, the biased prior $\beta^{senti.}_n$ enforces a seed word from *sentiment* topic *n* more probably to be generated from the *sentiment* topic *n*. The words distribution of each *aspect* topic $\phi^{aspect}_n \sim Dir(\beta^{aspect}_n)$ is similarly constructed.

For each word w in document x, a topic indicator λ is chosen from a topic class distribution p. The topic of the word w is then generated by z_w~$\text{Mult}(\theta^{(\lambda)})$, and the word itself is generated by w∈$\{1, \ldots, V\}$~$\text{Mult}(\phi^{(\lambda)}_{z_w})$. We summarize how the wsLDA model generates a corpus as follows (Dir and Mult mean Dirichlet and Multinomial distributions, respectively):

(1) For *sentiment* topic $n \in \{0, 1\}$:
  (a) Draw a multinomial word distribution over words: $\phi^{senti.}_n \sim \text{Dir}(\beta^{senti.}_n)$.
(2) For each *aspect* topic $k \in \{1, \ldots, K\}$:
  (a) Draw a multinomial word distribution over words: $\phi^{aspect}_k \sim \text{Dir}(\beta^{aspect}_k)$.
(3) For each *other* topic $m \in \{1, \ldots, M\}$:
  (a) Draw a multinomial distribution over words: $\phi^{other}_m \sim \text{Dir}(\beta^{other})$.
(4) For each document x in the corpus
  (a) Draw multinomial topic distributions $\theta^{senti.} \sim \text{Dir}(\alpha^{senti.})$, $\theta^{aspect} \sim \text{Dir}(\alpha^{aspect})$ and $\theta^{other} \sim \text{Dir}(\alpha^{other})$.
  (b) For each word of the document in $\{1,...,L_x\}$, where L_x is the length of document x:
    (i) Choose a topic class indicator $\lambda \sim \text{Mult}(p)$
    (ii) Choose a topic z_w from $\text{Mult}(\theta^{(\lambda)})$,
    (iii) Choose a word w from $\text{Mult}(\phi^{(\lambda)}_{z_w})$.
    (iv) Emit word w

In this work, we employ Gibbs sampling algorithm [9] to estimate the unknown parameters. We introduce some auxiliary notations. Let $\eta^{senti.}_{n,w}$ (or $\eta^{aspect}_{k,w}$, $\eta^{other}_{m,w}$) indicates the number of occurrences of *sentiment* topic *n* (or *aspect* topic *k*, *other* topic *m*) with word *w* in the whole corpus. All these counts are defined excluding the current word *w*. Then, by the property of Dirichlet distribution [9], we can easily compute the latent variables. The reader may refer to [9] for a detailed derivation of the sampling procedure. In this paper, we aim to obtain the word distributions of *sentiment* and *aspect* topics. Mathematically, we compute $\phi^{senti.}$ and ϕ^{aspect} as:

$$\phi^{senti.}_{n,w} = \frac{\beta^{senti.}_{n,w} + \eta^{senti.}_{n,w}}{\sum^{V}_{w'=1}\left(\beta^{senti.}_{n,w'} + \eta^{senti.}_{n,w'}\right)}; \quad \phi^{aspect}_{k,w} = \frac{\beta^{aspect}_{k,w} + \eta^{aspect}_{k,w}}{\sum^{V}_{w'=1}\left(\beta^{aspect}_{k,w'} + \eta^{aspect}_{k,w'}\right)} \tag{2}$$

We further transform the V-dimensional word distributions of *sentiment* topic i and *aspect* topic j to *sentiment* embedding $\boldsymbol{u}^{senti.}$ and *aspect* embedding $\boldsymbol{u}^{aspect}$ which are low-dimensional and have the same dimensions with the LSTM hidden states, by using a fully-connect layer:

$$\boldsymbol{u}_i^{senti.} = g\big(\boldsymbol{\phi}_i^{senti.}\boldsymbol{W}^{senti.}\big); \quad \boldsymbol{u}_j^{aspect} = g\big(\boldsymbol{\phi}_j^{aspect}\boldsymbol{W}^{aspect}\big) \tag{3}$$

where the matrices $\boldsymbol{W}^{senti.} \in \mathbb{R}^{V\times dim}$ and $\boldsymbol{W}^{aspect} \in \mathbb{R}^{V\times dim}$ are trainable parameters, *dim* is the size of the hidden states of our LSTM encoder, g is the non-linear function (i.e., tanh).

Soft Attention mechanism. Knowledge of sentiment lexicon and aspect plays a vital role in generating sentiment-aware review summary. We use lexicon embedding $\boldsymbol{u}^{senti.}$ to decide the lexicon-based attention weight for each word in text x, expecting that sentiment words obtain higher weights. Mathematically, the the representation of review x with lexicon-based attention weight (sentiment-aware review representation), denoted as $\boldsymbol{emb}_x^{senti.}$, takes the following form:

$$\boldsymbol{emb}_x^{senti.} = \sum_{t=1}^{N_x} a_t^{senti.}\boldsymbol{h}_t \tag{4}$$

$$a_t^{senti.} = \text{softmax}\big(\varrho(\boldsymbol{h}_t, \boldsymbol{u}^{senti.}; \Theta^{senti.})\big) \tag{5}$$

where N is the text length of the review, $\varrho(.;\Theta^{senti.})$ is a tanh activation function (has parameter set $\Theta^{senti.}$). The attention score $a_t^{senti.}$ indicates how likely the corresponding hidden state $\boldsymbol{h}_t$ acts as a sentiment indicter. In this way, the sentiment-intensive words in the review are supposed to contribute more to the review representation.

Similarly, the system should be sensitive to the domain-specific aspect constraints. Therefore, we use aspect embedding $\boldsymbol{u}^{aspect}$ to calculate the aspect-based attention score for each review x with given aspect. Similar to the calculation of $\boldsymbol{emb}_x^{senti.}$, the review representation with aspect-based attention weight (aspect-aware review representaiton), denoted by $\boldsymbol{emb}_x^{aspect}$, is computed as:

$$\boldsymbol{emb}_x^{aspect} = \sum_{t=1}^{N_x} \beta_t^{aspect}\boldsymbol{h}_t \tag{6}$$

$$\beta_t^{aspect} = \text{softmax}\big(\varrho(\boldsymbol{h}_t, \boldsymbol{u}^{aspect}; \Theta^{aspect})\big) \tag{7}$$

where Θ^{aspect} is the parameter set of the tanh function $\varrho(.;\Theta^{aspect})$.

Aspect/sentiment-aware Review Representation. To make the final representation of review x be aware of both aspect and sentiment knowledge, we combine aspect and lexicon based representations by additive projection:

$$\boldsymbol{emb}_x = \tanh\big(\xi\boldsymbol{W}^{f_s}\boldsymbol{emb}_x^{senti.} + (1-\xi)\boldsymbol{W}^{f_a}\boldsymbol{emb}_x^{aspect}\big) \tag{8}$$

where $\boldsymbol{W}^{f_s}$ and $\boldsymbol{W}^{f_a}$ are projection parameters. Parameter $\xi \in [0,1]$ is used to control the balance of effect of the aspect-aware and sentiment-aware representations, and is set to 0.5 in this work.

Domain Prediction. The final aspect/sentiment-aware review representation $\boldsymbol{emb}$ is fed into a task-specific fully-connected layer and a softmax layer for domain classification of the given review x:

$$y = \text{softmax}(\boldsymbol{V}_2 \cdot \boldsymbol{F}_x), \qquad \boldsymbol{F}_x = \tanh(\boldsymbol{V}_1 \cdot \boldsymbol{emb}_x) \tag{9}$$

where $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are projection parameters to be learned. The classifier is trained by minimizing the cross-entropy between the predicted distribution and the ground truth distribution.

3.2 Abstractive Review Summarization

The abstractive review summarization subtask shares the same review representation module with the domain classification subtask. Inspired by [11], the pointer-generator network is adopted as decoder to generate summaries. The pointer-generator network allows both copying words from input text via pointing (P_{vocab}), and generating words from a fixed vocabulary (P_{gen}). Thus, the pointer-generator has the ability to produce out-of-vocabulary (OOV) words.

On each step t, the decoder receives the word embedding of the previous word w_{t-1} and updates its hidden state $\boldsymbol{s}_t$. The attention mechanism is used to calculate the attention weights $\boldsymbol{a}_t$ and context vector $\boldsymbol{c}_t$ as in [11], which computed as a weighted sum of the hidden states of the input text. The context vector $\boldsymbol{c}_t$ is then concatenated with the decoder state $\boldsymbol{s}_t$ and fed through a linear layer and a *softmax* layer to produce the vocabulary distribution $P_{\text{vocab}}(w_t)$, which provides us with our final distribution from which to predict word w_t.

In the pointer-generator model, the generation probability $p_{\text{gen}} \in [0,1]$ for time step t is calculated from the context vector $\boldsymbol{c}_t$, the decoder state $\boldsymbol{s}_t$, and the decoder input $\boldsymbol{x}_t$ at time step t.

$$p_{\text{gen}} = \text{sigmoid}\big(\boldsymbol{U}_c^{\top}\boldsymbol{c}_t + \boldsymbol{U}_s^{\top}\boldsymbol{s}_t + \boldsymbol{U}_x^{\top}\boldsymbol{x}_t + b_{\text{gen}}\big) \tag{10}$$

where vectors $\boldsymbol{U}_c$, $\boldsymbol{U}_s$, $\boldsymbol{U}_x$ and scalar b_{gen} are learnable parameters.

For each review let the extended vocabulary be the union of the vocabulary and all words appearing in source review. We obtain the following probability distribution over the extended vocabulary:

$$P(w) = p_{\text{gen}}P_{\text{vocab}}(w) + (1 - p_{\text{gen}})\sum_{i:w_i=w} a_t^i \tag{11}$$

Once we have defined the summarization model, we can estimate the parameters to minimize the negative log-likelihood of the training data by using mini-batch stochastic gradient descent.

Overall, our *CASAS* model consists of two subtasks, each has a training objective. For the purpose of strengthening the learning of the shared aspect/sentiment review representations, we train these two related task simultaneously.

4 EXPERIMENTS

4.1 Datasets Description

We evaluate our model on Amazon reviews dataset from Stanford Network Analysis Project (SNAP) [5]. To test the performance of our model in cross-domain scenario, we use the 568,454 reviews from Food category as the source domain data and randomly choose 10,000 reviews from Electronics category as target domain data. Each review mainly contains product, user information, ratings, a plaintext review and a review summary.

4.2 Baseline Methods

In the experiments, we evaluate and compare our model with several baseline methods: the abstractive model (ABS) [7], the pointer-generator coverage networks (PGC) [11], the abstractive deep reinforced model (DeepRL) [8] (ML+RL version), the generative adversarial network for abstractive summarization (GANsum) [3].

Table 1: Out-of-domain experiments (on Electronics data).

Method	ROUGE-1	ROUGE-2	ROUGE-L	human eval.
ABS	54.32	35.83	50.95	2.05
PGC	57.64	37.82	53.15	3.24
DeepRL	58.42	37.93	54.26	3.05
GANsum	59.33	38.65	56.06	3.42
CASAS	**63.55**	**42.15**	**59.38**	**3.96**

Table 2: In-domain experiments (on Food data).

Method	ROUGE-1	ROUGE-2	ROUGE-L	human eval.
ABS	78.53	60.92	79.21	2.37
PGC	80.44	62.23	82.64	3.41
DeepRL	82.12	63.09	84.31	2.83
GANsum	82.48	63.65	84.13	3.49
CASAS	**83.25**	**64.45**	**85.98**	**3.77**

4.3 Implementation Details

We first settle down the implementation details for our weakly supervised LDA model. The numbers of *aspect* topic and *other* topic are set to 5 and 10 respectively. Other hyperparameters of wsLDA include: $\alpha^{senti.} = \alpha^{aspect} = 0.25$, $\alpha^{other} = 0.1$, β^{other}=0.1, γ_0=0.15, γ_1=0.85. We use 100-dimensional word2vec vectors pre-trained on English Wikipedia corpus to initialize the words in both datasets. The LSTM has hidden size 300. *CASAS* model is trained using Adadelta with mini-batch. Other hyperparameters include: learning rate 0.01, L2 regularization 0.001, dropout 0.2, batch size 64.

4.4 Experimental results

In this section, we compare our model with baseline methods from quantitative and qualitative perspectives.

4.4.1 Quantitative Evaluation. Following the same evaluation as in [7], we compare our model with baseline methods in terms of ROUGE-1, ROUGE-2, ROUGE-L and Human evaluation. We also perform human evaluation to evaluate the readability and quality of summaries. We randomly select 200 test examples from the dataset. For each example, three human evaluators are asked to rank each summary generated by all 5 models based on their readability, where 1 indicates the lowest level of readability while 5 indicates the highest level. We report the model comparison in both out-of-domain setup and in-domain setup.

Out-of-Domain Abstractive Review Summarization To test the performance of our model in cross-domain scenario, we train the proposed model on Food reviews data and test it on Electronics reviews data. Table 1 shows the performance of our *CASAS* model as well as the baseline methods. It clearly demonstrates that *CASAS* achieves better performance than the strong competitors (improves 7.1% on ROUGE-1, 9.1% on ROUGE-2, 6.0% on ROUGE-L, and 15.8% on Human evaluation).

In-Domain Abstractive Review Summarization We further report the experimental results in in-domain setup like literature [7]. Concretely, We randomly choose 1000 reviews from Food reviews dataset as test data, and the remaining reviews in Food reviews data are used for training. The experimental results are summarized

Table 3: Example summaries.

Input: this cable works FANTASTIC and i didnt have to pay $60 for it. i highly recommend!! so glad i gave a cheap price hdmi a shot. **Ground-truth:** Save yourself some money! **DeepRL:** Good item the cable. **CASAS:** Cable works fantastic, cheap price.
Input: I bought this for a 22TV for my son. I mounted this onto the corner stud and it works great! I love the angles it can achieve. Very sturdy and well built. It works perfect for my application. Very happy with this purchase! **Ground-truth:** High Quality/Low Price. **DeepRL:** Great angles loves. **CASAS:** Works great, very sturdy and well built

in Table 2. We observe that the proposed *CASAS* method substantially outperforms other methods and gets start-of-the-art on all evaluation metrics.

4.4.2 Qualitative Evaluation. To evaluate the proposed model qualitatively, we reported some generated summaries by different models. *CASAS* model is trained on Food reviews data and test it on Electronics reviews data. Due to the limitation of space, we randomly choose two generated summaries by DeepRL and our model from test data for comparison. The results are reported in Table 3. We observe that *CASAS* tends to generate more specific and meaningful summaries in response to the given texts. For example, our model successfully catches the "price" aspect of the cable and the corresponding sentiment word "cheap".

5 CONCLUSION

This work copes with a new problem in abstractive text summarization field – aspect/sentiment-aware abstractive review summarization in domain adaptation scenarios. Experiments show the superiority of *CASAS* in both out-of-domain and in-domain setups.

REFERENCES

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

[2] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997), 1735–1780.

[3] Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2018. Generative Adversarial Network for Abstractive Text Summarization. In *AAAI*.

[4] Duy Khang Ly, Kazunari Sugiyama, Ziheng Lin, and Min-Yen Kan. 2011. Product review summarization from a deeper perspective. In *Annual international ACM/IEEE joint conference on Digital libraries*. ACM, 311–314.

[5] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *ACM conference on Recommender systems*. ACM, 165–172.

[6] Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *ACL*. 339–348.

[7] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*.

[8] Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *arXiv preprint arXiv:1705.04304* (2017).

[9] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast collapsed gibbs sampling for latent dirichlet allocation. In *SIGKDD*. 569–577.

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

[11] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*.

[12] Min Yang, Dingju Zhu, M Rashed, and Kam-Pui Chow. 2014. Learning domain-specific sentiment lexicon with supervised sentiment-aware lda. *ECAI* (2014).