Adversarial Domain Adaptation Using Artificial Titles for Abstractive Title Generation

Francine Chen
FX Palo Alto Laboratory
Palo Alto, CA
chen@fxpal.com

Yan-Ying Chen
FX Palo Alto Laboratory
Palo Alto, CA
yanying@fxpal.com

Abstract

A common issue in training a deep learning, abstractive summarization model is lack of a large set of training summaries. This paper examines techniques for adapting from a labeled source domain to an unlabeled target domain in the context of an encoder-decoder model for text generation. In addition to adversarial domain adaptation (ADA), we introduce the use of artificial titles and sequential training to capture the grammatical style of the unlabeled target domain. Evaluation on adapting to/from news articles and Stack Exchange posts indicates that the use of these techniques can boost performance for both unsupervised adaptation as well as fine-tuning with limited target data.

1 Introduction

Many types of textual content, such as conversations and posts on chat, do not have a title or summary. While multi-sentence extractive summarization can give a sense of the content of an article, a title or highlight is more concise. Such short summaries can be generated using abstractive summarization with an RNN encoder-decoder model, e.g., (Nallapati et al., 2016).

A common issue when training models for abstractive summarization of conversations and posts is the lack of a large set of text with summaries. Obtaining good quality labeled data can be difficult and expensive, especially if author-generated summaries are desired. One option is to train on data from another domain with author-generated titles, but because of differences between domains, the performance may be less than adequate. These differences include different vocabularies, different grammatical styles, and different ways of expressing similar concepts. Vocabulary expansion may be used to address the different vocabularies in source and target domains, and adversarial domain adaptation (ADA) may be used to merge the embedded feature representations across domains. However, ADA does not adapt the decoder in an encoder-decoder generation model.

In this paper, we investigate the utility of these techniques in unsupervised domain adaptation for title generation. We also examine the use of a limited amount of labeled training data from the target domain, when high performance may be required but training data is not easily available. Our contributions include (1) proposing the use of artificial titles for unlabeled target documents to train a decoder to learn the grammatical style of titles in the new domain (2) proposing to train the decoder in a sequence of steps that encourages the source and target embedding spaces to remain aligned during adaptation, and (3) showing that our model improves performance over ADA and an expanded vocabulary alone and further, that a limited amount of labeled target data can achieve performance close to training on all labeled target data.

2 Related Work

Our model draws from work on abstractive summarization and unsupervised domain adaptation. Recently, a number of neural encoder-decoder models have been proposed for abstractive summarization e.g., (Rush et al., 2015; Chen et al., 2016a; Nallapati et al., 2016; Chopra et al., 2016; Li et al., 2017; Narayan et al., 2018; Hsu et al., 2018), with one of the better performing models being (See et al., 2017), which serves as our base model. Supervised domain adaptation methods have been proposed for generative models. (Hua and Wang, 2017) found that pre-training an abstractive summarizer with extractive summaries does not always improve performance, but (Chen et al., 2015) noted that fine-tuning a model trained

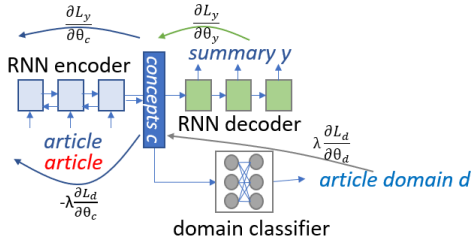

Figure 1: Encoder-decoder RNN model for text generation with a classifier for adversarial domain adaptation of the encoded representations (concepts) to an unlabeled target domain. Gradient reversal of L_d from the domain classifier to the encoder is indicated. The blue/red articles represent source/target domain data.

on source domain data with limited target domain data does improve performance.

A variety of techniques have been proposed for unsupervised domain adaptation of deep learning systems for classification, e.g., (Hsu et al., 2017; Tzeng et al., 2017; Ganin et al., 2016; Chen et al., 2016b; Ghifary et al., 2016). However, all used the aligned encoder representation for classification but not generation.

We adapt the domain-adversarial method for feature alignment in an encoder proposed by (Ganin et al., 2016). However, for text generation, a domain-independent representation from the encoder, as used in domain adaptation for classification, is not adequate. We also require the decoder to be adapted to varying domains to generate output appropriate for the target domain, an issue that we investigate in the context of title generation.

Jointly training a translation model with mixed *labeled* data from two domains can improve performance over training on one domain only (Pryzant et al., 2017). In contrast, our domain adaptation method trains sequentially on data, first with the *unlabeled* target domain data.

3 Domain-Adapted Title Generation

Our goal is to improve performance when labeled data from one domain, the *source*, is used to train a model which is then applied to another domain with no or only limited labeled data, the *target*.

3.1 Adversarial Domain Adaptation (ADA)

The embedded representation generated by the encoder, which represents the "concepts" in the input text, may differ across domains. To address this, we adapt the method proposed by (Ganin et al., 2016), which uses a domain classifier to force the concept representations to align across domains.

We use an encoder-decoder RNN model with domain adaptation (Figure 1) for title generation. Labeled source data is fed to the encoder and the decoder learns to generate summary titles. At the same time, the source data and unlabeled target domain data are encoded by a bidirectional LSTM as their concept representations, and the domain classifier tries to learn to differentiate between the representations of two domains.

The domain classifier has two dense, 100-unit hidden layers followed by a softmax. The concept representation vector is computed as the bidirectional LSTM encoder's final forward and backward hidden states concatenated into a single state. During training, the gradient from the domain classifier, $\frac{\partial L_d}{\partial \theta_d}$, is "reversed" to be negative before being propagated back through the encoder as $-\frac{\partial L_d}{\partial \theta_c}$, encouraging the embedded representations to align by adjusting the feature distributions to *maximize* the loss of the domain classifier.

In contrast to the two classification losses used by (Ganin et al., 2016) for training the model, we use the generated sequence loss together with the adversarial domain classifier loss:

$$loss = \frac{1}{T}\sum_{t=0}^{T} L_y(t) - \lambda L_d \quad (1)$$

where, following (See et al., 2017), the decoder (sequence) loss

$$L_y(t) = -logP(w_t^*) \quad (2)$$

is the negative log likelihood of the target word w_t^* at position t. The domain classifier loss, L_d, is the cross-entropy loss between the predicted and true domain label probabilities,

$$L_d = d \cdot logP(\hat{d}) + (1-d) \cdot log(1 - P(\hat{d})). \quad (3)$$

λ is a parameter relating the two losses.

We followed the schedule from (Ganin et al., 2016) for adjusting λ for the encoder:

$$\lambda_p = \frac{2}{1 + exp(-10p)} - 1 \quad (4)$$

λ was increased from 0.0 to 1.0 by increasing p from 0.0 to 1.0 over 5000 iterations, at which point we observed that the domain adaptation classifier loss was reaching an asymptote. λ was then held equal to 1.0 and training continued until validation performance for title generation reached an asymptote (when training on artificial titles or source data) or overtraining occurred (when training on limited target data). When updating the domain classifier, λ was set equal to one.

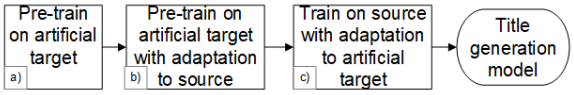

Figure 2: Flowchart for training a model for an unlabeled target domain with artificial targets.

3.2 Artificial Titles

The style of the unlabeled target may be different from the source, e.g., Stack Exchange is more casual and includes more slang than news articles. To capture the style of the unlabeled target, "artificial" titles were synthesized. Since titles tend to be short and encode-decoder models learn to model sentence length, target text between 4-10 words in length were selected. A common summary baseline is the first few sentences of a news article e.g. (Zajic et al., 2004; Nallapati et al., 2016); some social media sites, including Trip Advisor, Facebook and Reddit, display the first words of long posts. For example, this paragraph might be shown as "The style of the unlabeled target may ...".

The first text meeting the length requirement was selected 90% of the time and the second text meeting the requirement selected otherwise. For Stack Exchange, the text was a sentence from a post, and for news, where titles are often phrases, the text was a clause. Training on first text only, the loss dropped below 0.001 in less than 3k iterations, indicating the model had learned to copy from the first sentence. Use of the second text discourages this so that both the encoder and decoder are trained on text from the target domain (enabling use of an expanded, joint vocabulary trained on both source and target) to learn its style and vocabulary. However, the artificial titles will generally be different from the real titles, which may lead to lower summarization performance.

3.3 Sequential Training

Our adaptation method, **ASADA**, is shown in Figure 2: a) A model with a joint vocabulary is first pre-trained on artificial titles for the unlabeled target domain (Section 3.2). b) The embedding space of the pre-trained model is then adapted to the source domain using ADA (Section 3.1) to continue training on the target domain with the source domain as the auxiliary adaptation data. c) With a joint embedding space defined, the model is trained on the source domain, which has title-text pairs, and the unlabeled target domain is used as the auxiliary adaptation data to keep the model embedding aligned with the target data.

dataset	type	use	# train samples	summary length	
				mean	std dev
StackEx	artif.	T_{art}	398k	11.3	5.4
	filt-10	S,F	140k	6.5	1.4
News	artif.	T_{art}	287k	7.7	1.5
	filt-10	F	31k	9.0	1.4
	filt-14	S	168k	11.9	1.8

Table 1: Statistics of the Stack Exchange and News datasets. T_{art}: artificial Target; S: Source; F: fine-tuning; filt-X: filtered for at most length X.

4 Dataset

We used data from two domains: the public CNN/Dailymail (News) dataset used by (See et al., 2017) and posts from 20 Stack Exchange (StackEx) channels[1] with a bias towards those that are business related (see Appendix A for details). To reduce training time, each article was truncated to 200 words. We limited the data to those with title lengths of 10 words or less for use in fine-tuning because some were longer sentences rather than titles. (See Table 1) The News datasets were formatted as in (See et al., 2017). The StackEx dataset was randomly divided into train (90%), validation (5%) and test (5%).

5 Experiments

For all experiments, the Pointer-Generator model (Gulcehre et al., 2016) by (See et al., 2017) was used without coverage as our base model, since coverage is an additional training step that would add an additional variable to the comparisons. Although coverage improves performance by reducing repetitive words, we chose to examine the effects of different domain adaptation methods without it. For handling differences in vocabulary, the vocabulary of the labeled source and unlabeled target domains were combined. The union of the 50k most frequent terms from the training data of each domain produced a joint vocabulary of about 85k terms. When an individual vocabulary was used, the size was 50k words. When sequential training was used, a model was trained until the loss on a validation set reached an asymptote. Domain adaptation experiments from News to StackEx and from StackEx to News were conducted, first without target domain summary titles and then with a limited amount of target domain titles.

[1] https://archive.org/details/stackexchange downloaded 05/26/2017

id	reference or description	vocab	training data and method	News → StackEx ROUGE			StackEx → News ROUGE		
				1	2	L	1	2	L
(a)	See et al.	S	S	14.22	4.22	12.80	12.92	3.19	12.15
(b)	joint vocab	S+T	S	15.99	4.87	14.42	10.85	2.85	10.23
(c)	Ganin et al. (ADA)	S+T	S, S^{ADA}	16.75	5.24	15.10	12.45	3.12	11.53
(d)	artif titles	S+T	T_{art}	14.28	4.87	13.26	12.02	3.58	11.06
(e)	artif titles, ADA	S+T	T_{art}, S^{ADA}	16.88	5.35	15.24	14.36	3.84	13.47
(f)	ASADA	S+T	T_{art}, T_{art}^{ADA}, S^{ADA}	**17.78**	**6.22**	**16.15**	**16.75**	**6.11**	**15.99**
(g)	ASADA (lead-1)	S+T	T_{lead1}, T_{lead1}^{ADA}, S^{ADA}	16.46	5.30	15.01	16.16	3.36	14.64
(h)	Pryzant et al.(DM)	S+T	S+T_{art}	14.63	5.00	13.49	15.13	5.32	14.51
(i)	Pryzant et al. (ADM)	S+T	S+T_{art}	15.29	5.37	14.06	13.00	4.30	12.01
(j)	upper bound	T	T	31.49	13.70	29.22	23.52	10.92	22.34

Table 2: Title generation performance of domain adaptation from Source *S* to Target *T*. (a-c) Baselines. (d-g) Our approaches with artificial titles T_{art} and with lead-1 T_{lead1}, respectively. (h) DM: Discriminative Mixing. (i) ADM: Adversarial Discriminative Mixing. (j) Upper bound trained on labeled target data. Training steps are separated by commas. S^{ADA}: train on *S* using ADA. T_{art}^{ADA}: train on T_{art} using ADA.

id	prev training data & method	curr training data & method	domains gradually or jointly embedded?	same labeled data domain?
(E)	T_{art}	S^{ADA}	no	no
(F1)	T_{art}	T_{art}^{ADA}	yes	yes
(F2)	T_{art}^{ADA}	S^{ADA}	yes	no

Table 3: Comparison of adaptation steps with artificial titles using one step, (E), and two step ASADA, (F1) and (F2). (E) and (F) correspond to the models (e) and (f) in Table 2, respectively.

5.1 Unsupervised Target Domain Adaptation

For our investigations on domain adaptation when labeled target domain data is unavailable, models trained on source domain labels only and with a mix of source domain labels and artificial target labels are our baselines.

Effect of ADA and Vocabulary The top section of Table 2 shows baseline models trained
(a) with the source domain vocabulary [(See et al., 2017)'s approach without coverage]
(b) with a joint vocabulary instead of the source domain vocabulary
(c) model (b) followed by training using ADA to the target domain [(Ganin et al., 2016)'s approach].

The mixed results using a joint vocabulary reflect the better coverage of the added target words outside the source's top-50k vocabulary when the source is News vs. StackEx (see Appendix B). And when a joint vocabulary (S+T) is used, ADA (c) improves performance over training only on the source S (b), as expected.

Effect of Artificial Titles and Sequential Training The second section of Table 2 compares approaches using artificial titles:
(d) $\mathbf{T_{art}}$**:** a model pre-trained on target domain articles/posts with artificial target domain titles
(e) $\mathbf{T_{art}, S^{ADA}}$**:** model (d), further trained on the source with ADA to the target without labels.
(f) $\mathbf{T_{art}, T_{art}^{ADA}, S^{ADA}}$**:** ASADA. Model (d), followed by adapting the model, which has been trained on the target domain with non-optimal summaries, to source data, aligning the embedded representations of the two domains. Then the model is trained on source data with ADA to the unlabeled target to learn how to summarize while keeping the embedded representations aligned.
(g) ASADA using the lead-1 (first) sentence in place of T_{art}. The better performance in (f) supports ASADA'a use of artificial titles.

ASADA's two-step adaptation with artificial titles performed best out of all models. The mixed performance of training on T_{art} indicates the artificial title quality is lower for StackEx, (d) vs. (b). The weakly better performance of (e) over (c) indicates that applying S^{ADA} directly forgets much of T_{art}. The relative improvement of ASADA over training only on source was 25% (from News to StackEx) and 30% (from StackEx to News). This indicates that T_{art}^{ADA} allows the model to remember the vocabulary and style from T_{art} while learning how to summarize by S^{ADA}.

Table 3 illustrates differences between the one-step adaptation model (e), with id (E) and the two-step adaptation used in ASADA (F1 and F2). In both, the model is first trained on the target domain using T_{art}. In model (e), ADA then trains the encoder on source only and ignores T_{art}, gradually giving greater weight to the domain classifier, which uses the target data (see Sec. 3.1). At

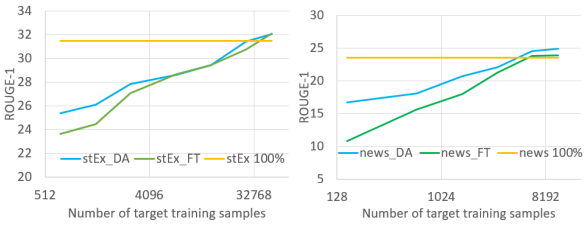

Figure 3: Domain adaptation performance with varying amounts of labeled StackEx (left) and News (right) data for fine-tuning with ADA (*_DA) and without (*_FT). For reference, performance when trained on all labeled target data and no adaptation (*_100%).

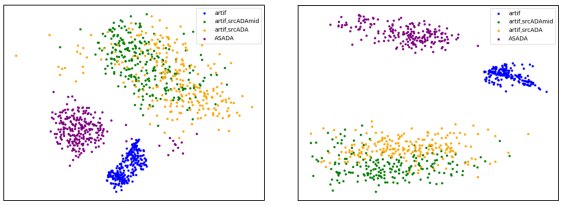

Figure 4: MDS visualizations comparing embeddings of a sample of test text produced by models (d), (e) and (f) in Table 2. artif: model (d). artif,srcADAmid: model (e) midway through ADA. artif,srcADA: trained model (e). ASADA: model (f). Left: News → StackEx. Right: StackEx → News.

the same time, the labeled data domain is switched to the source domain, so that both the embedding and decoder domains are abruptly changed. In contrast, in ASADA the embedding is gradually adapted from the target domain to jointly embed the source and target (F1). Only then is the target domain changed (F2).

In the third section, the labeled source is mixed with target domain artificial titles and trained using (Pryzant et al., 2017)'s Discriminative Mixed (DM) and Adversarial Discriminative Mixed (ADM) machine translation models. ADM is similar to ADA in that both use and adversarial classifier; however, for ADM both domains have labeled data. ASADA's better performance indicates that first pre-training with artificial titles to learn vocabulary and style and then adapting to the source to learn to summarize is better than jointly mixing artificial and true titles.

5.2 Limited Target Domain Labels

We next examine adaptation performance when a limited amount of labeled data is available for the target domain. Our best model for each domain, ASADA, is refined by training on various percentages of the labeled target domain training data and referred to as '*_DA' in Figure 3. For comparison, a baseline model was trained using labeled source domain data and then fine-tuned (Sun et al., 2016; Song et al., 2017) using labeled target domain data and is shown as '*_FT'.

Note that (1) when labeled target domain data is very limited, say 3,000 labeled samples, '*_DA' improves performance more than '*_FT' (2) as the amount of labeled target data increases, the performance with and without ADA increases, and with 30% of the target data (rightmost points) is close to or exceeds using 100% of the target data.

5.3 Visualization of Adaptation Models

Embedded points produced by models (d), (e) and (f) (see Section 5.1) are compared in the visualization in Figure 4. For the one-step adaptation model, (e), embedded points are shown partway through adaptation with ADA (i.e., p in Eqn. (4) is approximately 0.5) and after adaptation. The embedding partway through adaptation, labeled *artif,srcADAmid*, has moved away from the T_{art} embedding (model (d), labeled *artif*). After adaptation, labeled *artif,srcADA*, the embedded points are only slightly closer to the T_{art} embedded points. In contrast, the ASADA (f) embedding is closer to the T_{art} embedding and more compact, as is T_{art}. This supports our hypothesis that ASADA retains more of what was learned from the initial target embedding than model (e)'s one-step adaptation, contributing to ASADA's better performance.

6 Summary

We investigated unsupervised domain adaptation methods for an encoder-decoder model. We proposed the use of artificial titles for training a decoder to the target domain vocabulary and style and sequential adversarial domain adaptation to minimize rapid changes of the encoder embedding space. Our experiments show that our proposed approach performed best when compared to baseline adaptation techniques when unsupervised. And with very limited target domain labels for fine-tuning, our model performed better than fine-tuning a model trained on the source domain. In the future, we would like to understand the usefulness of artificial titles for training the decoder relative to other factors that may impact performance, e.g., how similar the true titles or summaries are in the different domains.

References

Qian Chen, Xiao-Dan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016a. Distraction-based neural networks for modeling document. In *IJCAI*, pages 2754–2760.

Xie Chen, Tian Tan, Xunying Liu, Pierre Lanchantin, Moquan Wan, Mark JF Gales, and Philip C Woodland. 2015. Recurrent neural network language model adaptation for multi-genre broadcast speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2016b. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. 2016. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 140–149.

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 *Proceedings of the 19th International Conference on Computational Linguistics (Long Papers)*, pages 132–141. Association for Computational Linguistics.

Wei-Ning Hsu, Yu Zhang, and James Glass. 2017. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *IEEE Automatic Speech Recognition and Understanding Workshop*.

Xinyu Hua and Lu Wang. 2017. A pilot study of domain adaptation effect for neural abstractive summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 100–106.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Reid Pryzant, Denny Britz, and Quoc Le. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126.

Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Xinhang Song, Luis Herranz, and Shuqiang Jiang. 2017. Depth cnns for rgb-d scene recognition: Learning from scratch better than transferring from rgb-cnns. In *AAAI*, pages 4271–4277.

Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of frustratingly easy domain adaptation. In *AAAI*, pages 2058–2065.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, page 4.

David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, pages 112–119.

A Stack Exchange Dataset

The Stack Exchange channels used for the dataset are: ai (i.e., ai.stackexchange.com), android, arduino, cs, datascience, emacs, engineering, freelancing, iot, opendata, opensource, patents, programmers, robotics, salesforce, sharepoint, travel, unix, webapps, and workplace.

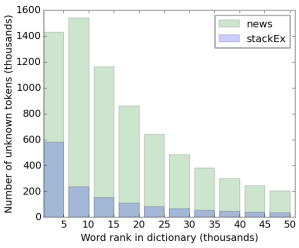

Figure 5: Histograms of News and Stack Exchange vocabularies showing the number of target domain joint vocabulary word tokens that are unrepresented in the source training data.

B Cross-Domain Vocabulary Coverage

For the expanded, joint vocabulary of source and target, Figure 5 shows that the number of News target tokens not represented by StackExchange vocabulary terms is much larger than the number of Stack Exchange target tokens not represented by News vocabulary terms. When trained on source only, these unrepresented target domain tokens are neither trained nor handled by the pointer-generator mechanism. Adversarial Domain Adaptation enables training of the *encoder* on these target tokens. Artificial Titles enable the *decoder* to be trained on these tokens.