Online Appendix

sDTM: A Supervised Bayesian Deep Topic Model for Text Analytics

Appendix A: Robustness Check: Attention Mechanism, RNN variants and Hidden Unit Size

Appendix B: Robustness Check: Learning Rate

Appendix C: Robustness Check: Optimizer

Appendix D: Robustness Check: Performance on Long Documents

Appendix E: Robustness Check: Incorporating Numerical Labels with sDTM

Appendix F: Topic Visualization and Topic Valence

Appendix G: The Relationship Between Individual Topic and Review Rating

Appendix A: Robustness Check: Attention Mechanism, RNN variants and Hidden Unit Size

An attention mechanism is an effective framework for learning representations of sequential text data. In a nutshell, an attention mechanism calculates different weights on each input unit in order to learn a weighted average of the input sequences. In our proposed sDTM approach, we modify the additive alignment attention mechanism (Bahdanau et al. 2015), as shown in Equation 1. In this appendix, we examine the effect of different attention mechanisms on the sDTM model performance, as measured by model fit perplexity. In particular, we implement two other popular attention mechanisms: the cosine attention (Graves et al. 2014) and the dot-product attention (Vaswani et al. 2017). The results are presented in Table A1. We can see that cosine attention achieves a higher perplexity than additive attention and the dot-product attention. This can be explained by the fact that cosine attention calculates a simple cosine similarity weight between input vectors and topic vectors, while our approach uses a fully-connected forward network to approximate the weight (Equation 1). Our approach, which uses additive attention, seems to achieve comparable performance with the approach that uses dot-product attention. In fact, the comparable performance outcomes of additive attention and dot product attention have been reported in prior studies (Vaswani et al. 2017).

Table A1: Model fitness (perplexity) using different attention mechanisms.

	Yelp		Stack Exchange	
	K=25	K =50	K=25	K=50
additive attention (our approach)	**664**	**669**	**696**	735
cosine attention	668	687	713	724
dot-product attention	670	680	701	**719**

For our recurrent neural network, we use Bi-GRU (bi-directional gated recurrent unit) due to its efficiency in modeling sequential text data. In its structure, GRU is similar to long short-term memory (LSTM) with a forget gate, but GRU has fewer parameters than LSTM, which makes the learning more efficient. To examine the effect of a recurrent neural network on sDTM text modeling performance, we choose several other standard recurrent neural network architectures, including GRU (single-directional), LSTM, Bi-LSTM, and a vanilla RNN network. The performance measured by model fit perplexity is shown in Table A2. We can see that sDTM is quite robust to the underlying recurrent neural network architectures. Bi-GRU and Bi-LSTM are two competitive models with comparable performance. This is expected because these two recurrent neural networks have very similar design, but Bi-GRU is more efficient in training due to fewer parameters. The vanilla RNN model performs the worst of the models evaluated and takes significantly longer.

Table A2: Model fitness (perplexity) using different recurrent neural network models.

	Yelp		Stack Exchange	
	K=25	K =50	K=25	K=50
Bi-GRU (our approach)	**664**	**669**	**696**	735
GRU	671	678	703	751
Bi-LSTM	670	673	698	**730**
LSTM	690	697	715	749
RNN	704	715	726	768

Lastly, we examine the effect of hidden unit size in the recurrent neural network GRU. In our implementation, we choose a moderate hidden size of 64. We vary the unit size to 32, 128 and 256 and evaluate the model fitness performance. The results are presented in Table A3. We can see that increasing unit size may further decrease model perplexity, but it leads to more model parameters to optimize. Therefore, a moderate size of 64 is a reasonable unit size for sDTM in both datasets.

Table A3: Model fitness (perplexity) using different hidden unit sizes in GRU.

	Yelp		Stack Exchange	
Unit Size	K=25	K =50	K=25	K=50
32	703	756	736	754
64	664	669	696	**735**
128	672	668	712	743
256	**657**	**656**	**685**	682

Appendix B: Robustness Check: Learning Rate

Learning rate is an important parameter for effective model optimization. It controls how much to adjust the model weights with respect to the loss gradient. A high learning rate will make the learning jump over minimum and lead to failed training, and a low learning rate will require more training epochs (more resources) or may even get stuck in an undesirable local minimum. To understand how learning rate affects sDTM training, we choose four learning rates (0.01, 0.005, 0.001 and 0.0001) and plot the model fit perplexity on the training set and validation set in Figure B1. We train the model on the Yelp dataset with 50 topics. Recall that perplexity is a function of data likelihood, and lower perplexity indicates higher likelihood. We can see that, as expected, when we set the learning rate to a high value (0.01 and 0.005), even though the training perplexity becomes converged, the validation perplexity becomes infinitely large and cannot be correctly calculated. When we set learning rate to be as small as 0.0001, the model converges slowly and still does not reach optimal points even after 200 epochs. The learning rate is better set at a moderate value of 0.001 because we can see that the model training loss quickly converges and the validation loss increases after around 75 epochs, indicating that a desirable model (one with neither overfitting nor underfitting concerns) is obtained. We have tested the learning rate on other datasets and topic settings and find 0.001 is a reasonable learning rate. For future work that trains sDTM, we recommend using 0.001 as the starting learning rate and plotting the training/validation loss for further monitoring.

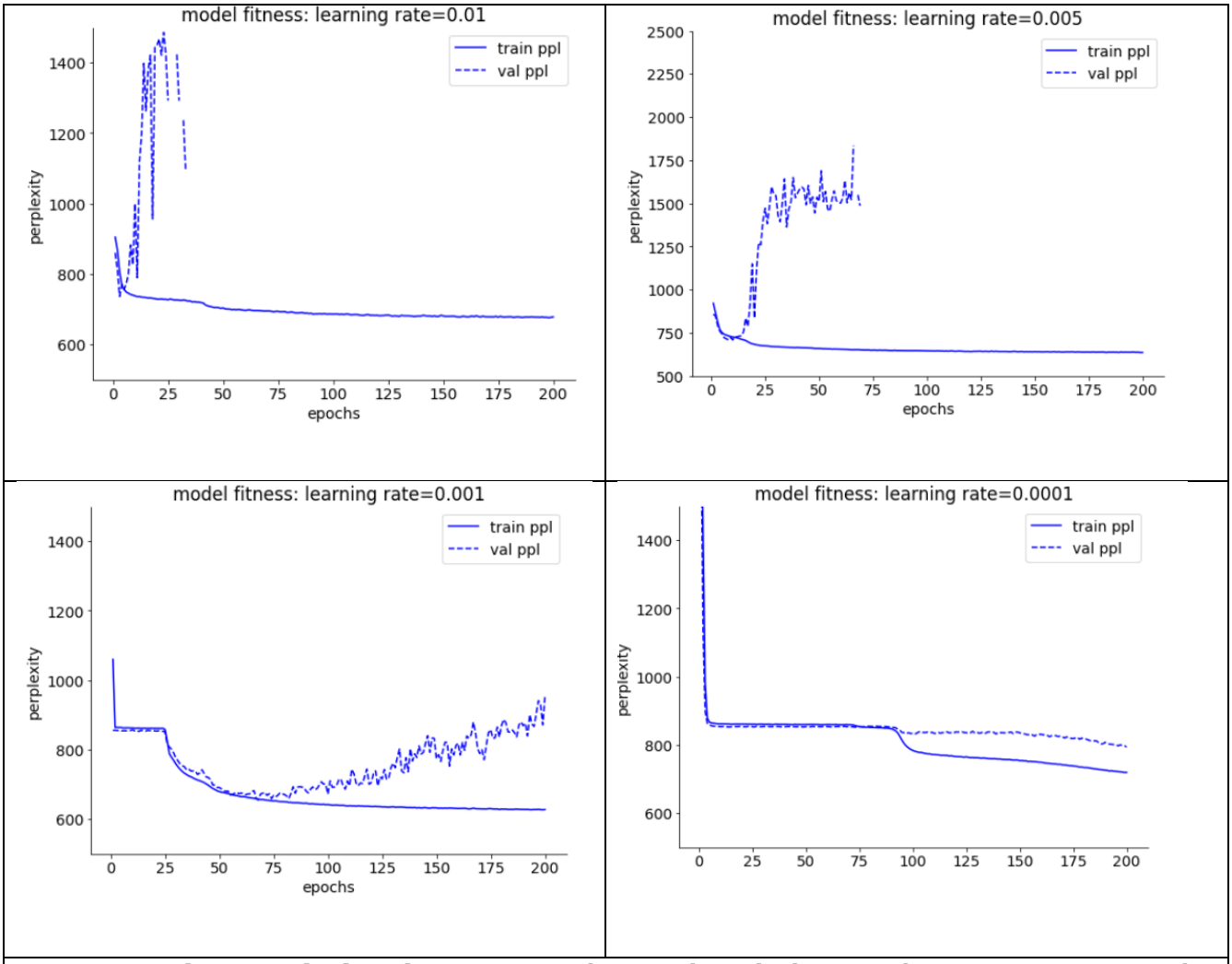

Figure B1: Robustness check on learning rate. This uses the Yelp dataset with 50 topics. We can see that the learning rate of 0.001 has the overall best performance in terms of convergence and training efficiency. For the top two figures (learning rates =0.01 and 0.005), the validation perplexity becomes infinitely large and cannot be correctly calculated after around 50 epochs.

Appendix C: Robustness Check: Optimizer

We use stochastic gradient descent as the optimization algorithm for training sDTM. In practice, researchers may need to choose a particular optimizer in the stochastic gradient descent. We investigate the effect of different learning optimizers on sDTM training performance. We choose four commonly used optimizers for comparison: Adam, RMSProp, gradient descent (with momentum), and Adagrad optimizer. The performance is shown in Figure C1. We can see that the Adam optimizer has the overall best performance over others and it converges to the lowest perplexity on the validation set. In observing

the gradient descent momentum and the Adagrad optimizer, we see that although in both cases the loss converges relatively faster than it does with the Adam optimizer, they both get stuck at a perplexity much higher than that of the Adam optimizer, which may indicate a possible local minimum. For the RMSProp optimizer, the fast increase in validation perplexity indicates a severe overfitting. The model fit performance of different optimizers is similar to that obtained in other datasets and topic settings, and we thus recommend using the Adam optimizer in training sDTM.

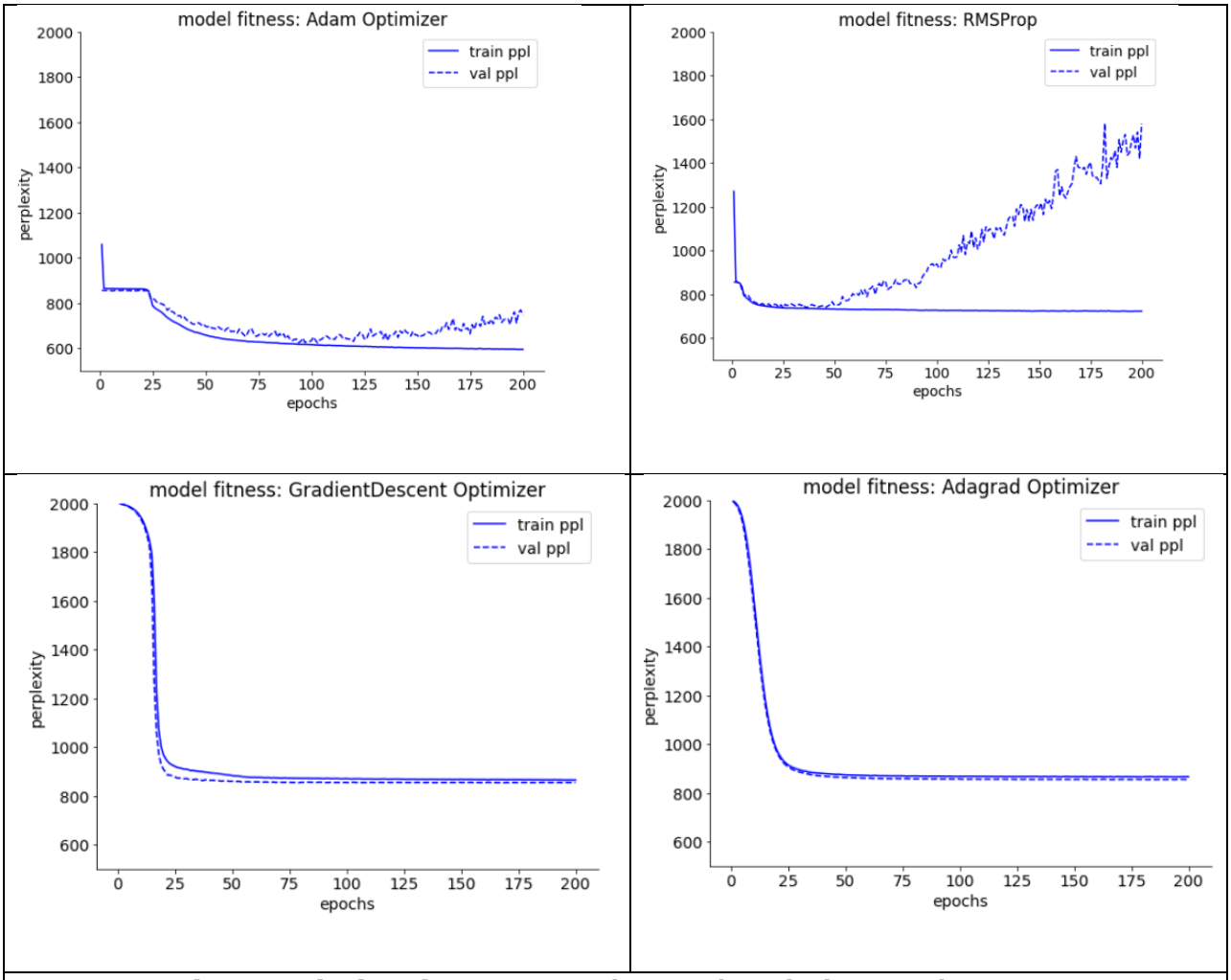

Figure C1: Robustness check on learning rate. This uses the Yelp dataset with 50 topics. We can see that the Adam optimizer has the overall best performance in terms of convergence and training efficiency.

Appendix D: Robustness Check: Performance on Long Documents

In our study, we consider commonly used textual datasets of consumer reviews and online knowledge community Q&As. The average text length within the Yelp review and Stack Exchange datasets are 110 and 173, respectively. How does sDTM perform on a text dataset with significantly longer text? We examine the generalizability of our approach on longer text. In particular, we consider a dataset of Wikipedia articles (Zubiaga 2012). The dataset contains 18,524 Wikipedia articles with an average length of 1,020 words, which is significantly longer than the Yelp and Stack Exchange datasets. In the Wikipedia dataset, each article is associated with multiple labels such as architecture, environment, TV, etc. There are 73 labels in total. Therefore, the prediction task is a multi-label classification. For this dataset, we use a sequence length of 1,500 for RNN and a hidden size of 256 for GRU. We compare sDTM with LDA and the neural topic model NTM with respect to model fitness and prediction accuracy. The results are presented in Table D1, and they further validate that our approach, sDTM, outperforms LDA and NTM in both text modeling and predictive power when used for long documents. sDTM can handle long documents very well for two reasons. First, at its core, sDTM is a dimension reduction method that reduces the high-dimensional text data into a low-dimensional topic space. Long documents exhibiting more topic variety may even help topic modeling (Tang et al. 2014). Second, we use an attention mechanism to get a weighted representation of the input sequence. Unlike vanilla RNN, which may suffer from dependency issues when used with long documents, an attention mechanism can overcome the long sequence problem (Bahdanau et al. 2015). Therefore, these two key design choices allow sDTM to generalize for long documents.

Table D1: Model performance on Wikipedia dataset. Model fit perplexity and multi-label classification accuracy F1-score are reported.

	Model Fitness (perplexity)		Multi-Label Classification (F1-score)	
	K=25	K=50	K=25	K=50
LDA	1291	1137	0.231	0.261
NTM	929	911	0.172	0.241
sDTM	**918**	**899**	**0.517**	**0.519**

Appendix E: Robustness Check: Incorporating Numerical Labels with sDTM

Our proposed approach, sDTM, can also incorporate numerical labels as auxiliary labels. To demonstrate its effectiveness in incorporating numerical labels, we conduct experiments on a movie review dataset. The Movie Review Data (MRD) (Pang and Lee 2005) is a collection of movie reviews with scaled sentiment scores ranging from 0 to 1. The dataset contains 5,006 movie reviews with an average length of 188 words. Supervised learning for the movie review data is a regression task because the label is numeric. Following Mcauliffe and Blei (2008), we evaluate the quality of regression using the "predictive R^2 (pR^2)." pR^2 is defined as the fraction of variability in the out-of-sample response variable, which is captured by the out-of-sample prediction $pR^2 = 1 - (\sum(z - \hat{z})^2)/(\sum(z - \bar{z})^2)$, where z is true value, $\hat{z}$ is predicted value, and $\bar{z}$ is mean of out-of-sample response variable. The model performance on the MRD is presented in Table E1. We see that sDTM consistently outperforms LDA and NTM significantly, since LDA and NTM struggle to learn valence-oriented topics. On the contrary, similar to what we observed in the experiments on the Yelp dataset, sDTM, leveraging the supervised rating information, can learn more valence-oriented topics and thus shows superior predictive performance.

Table E1: Model performance on MRD. Model fit perplexity and multi-label classification accuracy F1-score are reported.

	Model Fitness (perplexity)		Regression (pR^2)	
	K=25	K=50	K=25	K=50
LDA	1415	1714	0.174	0.142
NTM	919	915	0.152	0.144
sDTM	**915**	**902**	**0.598**	**0.602**

Appendix F: Topic Visualization and Topic Valence

In Table 6 and Table 7, we present the topics learned by LDA and our approach, sDTM. The comparison highlights the fact that sDTM leverages sentiment labels and is capable of learning more coherent, distinguished and valence-oriented topics. In this appendix, we present the topics learned by the neural topic model, NTM, which is the major component of our sDTM approach, in Table F2. The valence score

of each topic is calculated as the weighted sum of the most probable words' VADER sentiment scores. We can clearly see from Table F1 that the topics, similar to those learned by LDA, are devoid of valence. This is because NTM is still an unsupervised topic model approach that ignores the valuable sentiment labels accompanying the consumer reviews.

As a robustness check, following prior studies (Abbasi et al. 2011; Ahmad et al. 2020), we use another popular sentiment lexicon, SentiWordNet (Baccianella et al. 2010), to measure the topic valence. Since SentiWordNet provides positive and negative sentiment score for each sense of a word, we calculate a word sentiment score as the difference between the total positive score and total negative score. We tabulate the topic valence distribution statistics of LDA, NTM and sDTM in Table F1. We observe that similar to its results using the VADER lexicon, sDTM is capable of learning topics with even more significant sentiment valence, as indicated by more negative and positive valence and higher valence standard deviation. Given that IS researchers expect to use topic modeling to derive meaningful and coherent topics and identify topic valences, we recommend that researchers incorporate labels to enhance topic modeling for subsequent empirical and predictive analytics.

Table F1: Topic valence statistics of different models. Topic valence score is calculated as the weighted sum of the top 50, top 1,000 (half of the vocabulary), and top 2,000 (all vocabulary) words' SentiWordNet sentiment scores.

	N=50				N=1,000				N=2,000			
	min	max	mean	stdev	min	max	mean	stdev	min	max	mean	stdev
LDA	-2.52	70.83	3.13	14.77	11.13	91.46	51.08	25.92	17.63	91.38	54.49	22.54
GSM	-0.67	52.71	3.30	11.75	0.46	77.80	46.92	22.50	0.37	77.64	49.29	20.27
sDTM	-8.77	129.39	12.05	34.20	-3.72	153.87	53.79	41.82	-5.20	153.64	61.36	35.92

Topic	Polarity score	NTM: Most probable words									
1	0	teacher	instructor	guitar	amenities	eddie	trainer	trail	bike	sink	reclining
2	0	honda	salon	appointment	repair	warranty	gel	pedicure	maid	manicure	technician
3	0	chicken	salad	rib	potato	dish	vegetarian	greek	soup	sushi	entrees
4	2.527	steak	menu	delicious	bread	dessert	meal	taste	canoe	taco	tasty
5	0	ruth	fajita	taiwanese	scrambler	undercook	berry	quesadilla	ham	raspberry	pepperoni
6	0.247	gym	shoes	desk	class	security	sales	facility	bed	performance	shower
7	1.664	donuts	coffee	game	bakery	bartender	crowd	sport	snack	seat	fun
8	0	drago	espresso	morning	bosa	barista	est	que	study	frosting	team
9	0	food	table	chocolate	server	waitress	eat	restaurant	order	tea	milk
10	0	drago	bosa	barista	starbucks	espresso	doughnut	que	est	bagel	rio
11	0	hotel	room	king	class	ticket	magic	store	mac	clothes	bed
12	0	burger	truffle	onion	spicy	pork	fries	noodle	juicy	gordon	broth
13	0	waitress	chocolate	cream	tea	server	pizza	patio	food	milk	eat
14	0	salsa	mexican	brunch	sandwich	taco	pizza	bake	strawberry	cream	patio
15	0.727	golf	lounge	bay	club	topgolf	entertainment	venue	golfer	blast	mall
16	0	cupcake	latte	cookies	aunt	food	beer	hostess	cup	tre	frosting
17	0	patio	mexican	boba	salsa	bake	chocolate	bubble	pastries	sugar	madison
18	2.471	steak	dessert	bread	menu	vegan	delicious	portion	canoe	salad	appetizer
19	1.872	game	fun	snack	donuts	crowd	variety	sport	seat	joe	coffee
20	0.724	cake	drink	beer	coffee	bar	bakery	bartender	seat	game	party
21	0	pastry	croissant	breakfast	boba	patio	milk	frosting	mexican	fruit	madison
22	0	coffee	drink	cake	bar	bakery	beer	seat	game	bartender	cup
23	0	breakfast	patio	fruit	pastries	cupcake	boba	frosting	tea	milk	café
24	2.802	trader	fun	joe	snack	play	ball	outdoor	game	donuts	grocery
25	0	hotel	ticket	movie	mac	theater	outlet	mall	store	magic	room

Table F2: 25 topics in the Yelp review dataset learned by NTM. Topic valence score is calculated as the weighted sum of the most probable words' VADER sentiment scores.

Appendix G: The Relationship Between Individual Topic and Review Rating

Table 6, Table 7 and Table F2 have shown that sDTM is capable of learning topics encompassing valence, while LDA's topics are more likely to be devoid of valence. Prior researchers are interested in identifying topics that are related to sentiment, i.e., topic valence. For instance, applying LDA to consumer reviews, Tirunillai and Tellis (2014) identify positive and negative topics related to mobile phones in dimensions such as compatibility (positive) and discomfort (negative). Therefore, we expect that learning topics encompassing valence can facilitate effective text data exploration and help researchers accurately pin down key topics of interest in subsequent studies. In this appendix, we further show that the topics encompassing valence are useful in downstream analysis.

We infer topic distributions of Yelp reviews in the held-out test set using LDA and sDTM, with each learning 25 topics. It is worth noting that sDTM does not require labeling during inference. Rather, it automatically infers a review's topic distribution using the bag-of-words input, similar to LDA. Following Yang et al. (2018), we code a review as positive if it gives a rating of four or five starts, and as negative with three stars or below. Thus, the dependent variable is the coded binary rating: *sentiment*. The independent variable is each review's topic probability, $Topic_k$, for a given topic k. This empirical study examines the relationship between the review sentiment and individual topics, i.e., whether the reviewers tend to give a positive/negative rating when a certain topic is strongly/weakly mentioned. We also include several control variables including (1) *sequence*, the number of reviews posted before the focal review, and (2) *words*, the number of words in the review. For each individual topic, we estimate the following logit model:

$$logit(sentiment) = \beta_0 + \beta_1 Topic_k + \beta_2 Sequence + \beta_3 \log(words) + \epsilon,$$

Since there are 25 topics, we estimate 25 models each for LDA and sDTM. The main regression results of the LDA model and the sDTM model are presented in Table G1 and Table G2, respectively. We only present the regression models where the topic independent variable is negatively correlated with

sentiment. A comparison between the two regression results leads to several findings. First, for those LDA topics that are significantly and negatively associated with the review sentiment, some do not show any negativity based on the most probable words (see Table 6). For example, regression results show that topic 3 is negatively associated with review rating ($\beta = -1.208$), indicating that reviewers tend to give a negative rating if they discuss topic 3 in the reviews. By examining the key words identified by LDA in topic 3 (lounge, flight, popcorn, airport, priority, gate) in Table 6, we can ascertain only that this is a topic related to an airport lounge; the topic valence is not clear. This inconsistent result does not help explain why topic 3 has a negative impact on sentiment. We can see that using LDA, researchers who would like to find negative topics without running regression analyses are more likely to miss them. In contrast, the main regression results of sDTM (Table G2) show that the negative topics identified (Table 7) are all negatively and statistically significantly correlated with the review sentiment. For example, the most probable words (apology, horrible, argue, ignore, disrespectful, rude, mistake) in topic 3 in sDTM indicate a strong negative topic related to service quality, and its effect on review sentiment is -4.965 (*p-value=0.000*). Second, we can observe that the effect size of sDTM topics is substantially higher than the effect size of LDA topics. The untabular regression results of NTM are similar to those of LDA. Putting together this appendix and results in Section 5, we can conclude that the proposed approach sDTM can leverage review's rating label and extract more coherent topics encompassing valence, which benefits researchers' data exploration and empirical analysis.

Table G1: Regression results of the effect of review topic inferred by LDA on review sentiment.

DV: Sentiment; logit model; LDA; N=8,065								
Intercept	3.155***	3.024***	3.236***	3.034***	3.107***	3.110***	3.171***	3.080***
	(0.151)	(0.153)	(0.152)	(0.158)	(0.150)	(0.150)	(0.152)	(0.151)
Topic3	-1.208***							
	(0.303)							
Topic4		-1.965***						
		(0.122)						
Topic7			-2.606***					
			(0.253)					
Topic12				-4.531***				
				(0.185)				
Topic13					-0.307***			
					(0.118)			
Topic15						-2.011***		
						(0.317)		
Topic16							-2.419***	
							(0.215)	
Topic25								-2.034***
								(0.204)
words	-0.555***	-0.485***	-0.564***	-0.473***	-0.544***	-0.541***	-0.546***	-0.527***
	(0.032)	(0.033)	(0.033)	(0.034)	(0.033)	(0.032)	(0.033)	(0.033)
sequence	0.000*	-0.000	0.000*	0.000***	0.000*	0.000*	0.000*	0.000
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

Notes. Standard errors in parentheses. * p<.1, ** p<.05, ***p<.01

Table G2: Regression results of the effect of review topic inferred by sDTM on review sentiment.

DV: Sentiment; logit model; sDTM; N=8,065							
Intercept	3.255***	3.432***	3.472***	3.327***	3.322***	3.084***	3.122***
	(0.155)	(0.158)	(0.156)	(0.154)	(0.153)	(0.154)	(0.150)
Topic1	-4.965***						
	(1.303)						
Topic3		-8.790***					
		(0.418)					

Topic8			-13.355***				
			(0.692)				
Topic13				-7.653***			
				(0.500)			
Topic16					-7.248***		
					(0.897)		
Topic20						-3.204***	
						(0.180)	
Topic24							-0.317*
							(0.180)
words	-0.562***	-0.545***	-0.549***	-0.538***	-0.559***	-0.479***	-0.550***
	(0.033)	(0.034)	(0.033)	(0.033)	(0.033)	(0.033)	(0.032)
sequence	0.000	0.000***	0.000**	0.000***	0.000	-0.000	0.000**
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

Notes. Standard errors in parentheses. * p<.1, ** p<.05, ***p<.01

REFERENCES

Abbasi, A., France, S., Zhang, Z., and Chen, H. 2011. "Selecting Attributes for Sentiment Classification Using Feature Relation Networks," *IEEE Transactions on Knowledge and Data Engineering* (23:3), pp. 447–462.

Ahmad, F., Abbasi, A., Li, J., Dobolyi, D. G., Netemeyer, R. G., Clifford, G. D., and Chen, H. 2020. "A Deep Learning Architecture for Psychometric Natural Language Processing," *ACM Transactions on Information Systems* (38:1), pp. 1–29.

Baccianella, S., Esuli, A., and Sebastiani, F. 2010. "Sentiwordnet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.," in *Lrec* (Vol. 10), pp. 2200–2204.

Bahdanau, D., Cho, K., and Bengio, Y. 2015. "Neural Machine Translation by Jointly Learning to Align and Translate," in *International Conference on Learning Representations*.

Graves, A., Wayne, G., and Danihelka, I. 2014. "Neural Turing Machines," *ArXiv:1410.5401 [Cs]*.

Mcauliffe, J. D., and Blei, D. M. 2008. "Supervised Topic Models," in *Advances in Neural Information Processing Systems 20*, pp. 121–128.

Pang, B., and Lee, L. 2005. “Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales,” in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 115–124.

Tang, J., Meng, Z., Nguyen, X., Mei, Q., and Zhang, M. 2014. “Understanding the Limiting Factors of Topic Modeling via Posterior Contraction Analysis,” *International Conference on Machine Learning*, pp. 190–198.

Tirunillai, S., and Tellis, G. J. 2014. “Mining Marketing Meaning from Online Chatter: Strategic Brand Analysis of Big Data Using Latent Dirichlet Allocation,” *Journal of Marketing Research* (51:4), pp. 463–479.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. 2017. “Attention Is All You Need,” in *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Yang, M., Adomavicius, G., Burtch, G., and Ren, Y. 2018. “Mind the Gap: Accounting for Measurement Error and Misclassification in Variables Generated via Data Mining,” *Information Systems Research* (29:1), pp. 4–24.

Zubiaga, A. 2012. “Enhancing Navigation on Wikipedia with Social Tags,” *ArXiv:1202.5469 [Cs]*.