Bagging Model for Product Title Quality with Noise[*]

CIKM AnalytiCup 2017[†]

Tam T. Nguyen[‡]
Ryerson University
nthanhtam@gmail.com

Hossein Fani[‡]
University of New Brunswick
hosseinfani@gmail.com

Ebrahim Bagheri[‡]
Ryerson University
ebrahim.bagheri@gmail.com

Gilberto Titericz
Airbnb, Inc.
giba1978@gmail.com

ABSTRACT

We present our winning approach for the Lazada Product Title Quality Challenge for the CIKM Cup 2017 where the data set was annotated as *conciseness* and *clarity* by Lazada QA staffs. The participants were asked to build machine learning model to predict *conciseness* and *clarity* of an SKU based on product title, short description, product categories, price, country, and product type. As sellers could freely enter anything for title and description, they might contain typos or misspelling words. Moreover, there were many annotators labelling the data so there must be disagreement on the true label of an SKU. This makes the problem difficult to solve if one is solely using traditional natural language processing and machine learning techniques. In our proposed approach, we adapted text mining and machine learning methods which take into account both feature and label noises. Specifically, we are using bagging methods to deal with label noise where the whole training data cannot be used to build our models. Moreover, we think that for each SKU, *conciseness* and *clarity* would be annotated by the same QA. It means that *conciseness* and *clarity* should be co-related in a certain manner. Therefore, we extended our bagging approach by considering out of fold leakage to take advantage of co-relation information. Our proposed approach achieved the root mean squared error (RMSE) of 0.3294 and 0.2417 on the test data for *conciseness* and *clarity*, respectively.

KEYWORDS

Product Title Quality, E-Commerce, CIKM AnalytiCup

ACM Reference format:
Tam T. Nguyen, Hossein Fani, Ebrahim Bagheri, and Gilberto Titericz. 2017. Bagging Model for Product Title Quality with Noise. In *Proceedings of International Conference on Information and Knowledge Management, Pan Pacific Singapore, 6-10 Nov. 2017 (CIKM'17),* 4 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Customers do indeed fall prey to cognitive bias of 'judging a book by its cover' and product's title functions as the sole factor influencing their decision of whether to buy or skip a particular product. As a result, titling of a product becomes essential in e-commerce for sellers. Sellers attempt a clear, yet concise title which not only describes a product and its features comprehensively, but also stands out the product among all the likes in the result of product search engines for customers. However, seeking customer's attention, some inexperienced or malicious sellers might choose inaccurate or fraudulent titles such as *'Hot Sexy Tom Clovers Womens Mens Classy Look Cool Simple Style Casual Canvas Crossbody Messenger Bag Handbag Fashion Bag Tote Handbag Gray'*.

In the Lazada product title quality challenge, given a set of product titles, description, and attributes, we aim to build a product title quality model that can assess the *clarity* and the *conciseness* of a product title systematically and automatically, concluding to a state of clean and relevant product titles. Specifically, we have defined two independent neural-based regressors to rate the quality measures, clarity and conciseness, based on the feature set including n-gram character of the product title and other features such as title length, description length, number count.

2 DATASET

We evaluate our proposed model on a data set of more than 60K products from Lazada, southeast Asia's number one online shopping and selling destination. Besides its title, each product is accompanied by id, three-level hierarchical category, description, price, country of origin, delivery scope.

Additionally, the product title's clarity and conciseness are measured manually by Lazada's internal quality control team based on the following guidelines:

- **clarity:** The product title is clear if within five seconds one can understand the title, what the product is, and quickly figure out the key attributes (color, size, model, ...).
- **conciseness:** The product title is concise if it is short enough to contain all the necessary information. Otherwise, i.e., the

[*] The source code is available at https://goo.gl/1dmsXZ
[†] https://competitions.codalab.org/competitions/16652
[‡] Laboratory for Systems, Software and Semantics, Ryerson University

CIKM'17, 6-10 Nov. 2017, Pan Pacific Singapore
© 2017 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1: Disagreement in target functions clarity and conciseness for similar titles.

sku_id	title	clarity	conciseness
VO749FAAAAD86YANMY	*1 Pair of Unisex Touch Screen Sensitive Gloves Knitted Winter Warm Christmas Glove Coffee*	0	1
VO749FAAAAD7E8ANMY	*1 Pair of Unisex Touch Screen Sensitive Gloves Knitted Winter Warm Christmas Glove Red*	0	0

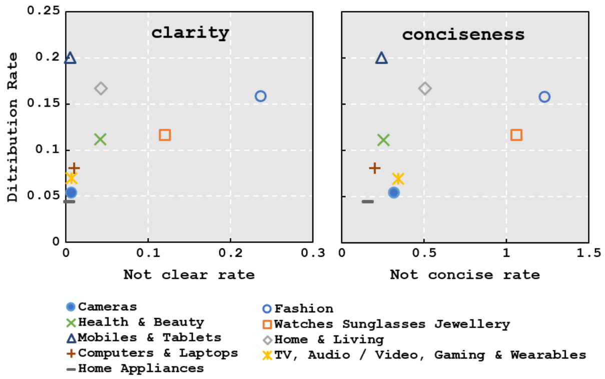

Figure 1: Distribution of product general categories (category_lvl_1) vs. not clear and not concise target functions.

title is too long with many unnecessary words, Or it is too short such that it is unsure what the product is.

2.1 Data Exploration

We first explore the data at each product instance including clarity and conciseness labels. We found that the dataset has outliers in 'price' (-1, 999999, 9999999). More importantly, we discover that there are some disagreement in titles' clarity and conciseness such that very similar titles receive different judgments as in Table 1.

Secondly, we inspect each property to find more pieces of information. We found that the first 2 characters in 'sku_id' and product brand name are often but not always the same. Brand names also happen to be the first term of 'title'.

Thirdly, we also check the correlation of the two labels. Interestingly, we found that there are only three combinations for the pair. While titles which are clear might be concise or not, there are no title which is not clear, yet concise. Intuitively, if a title is not clear, it is not concise either.

Finally, we study the distribution of products over clarity and conciseness based on different categories as shown in Figure 1. Compared to other categories, Fashion and Watches categories is quite balanced that makes prediction problem easier on these categories.

3 PROPOSED APPROACH

In this section we describe our proposed approach including feature engineering and modelling.

3.1 Feature Engineering

Before extracting features, we preprocess the data by removing HTML tags, removing special characters, and replacing missing values.

Table 2: All features set.

group	features	name
statistics	#char(length)	
	#term	xg_feat
	price	price_feat
information	color	color
	brand	brand
	category entropy	entropy_feat
n-gram term	(1,3)-gram	bow.3grams
n-gram char	(1,6)-gram	boc.6grams
	#upper char	
	#special char	
	html escape	
	#invisible char	
sparse feature	category one-hot-encoding	sp_feat
leave-one-out encode		
embedding	word2vec[2]	
part-of-speech	#adjective	
	#verb	
	#noun	
	#number	
multilingual characters	#non-english char	
	#chinese char	char_set_feat

Table 3: Most important features based on linear SVM.

label	name	coef.	label	name	coef.
clarity	t	0.442989	conciseness	my	1.087967
	sexy	0.398171		ph	0.968527
	exy	0.398026		c	0.957356
	sex	0.384535		ocal	0.931618
	urse	0.368735		local	0.925727
	purse	0.341463		r	0.920576
	purs	0.341463		loca	0.912073
	rse	0.338007		sg	0.909163
	xy	0.334105		cal	0.888805
	purse	0.326108		loc	0.882074

3.1.1 Feature Extraction. Besides the non-textual attributes of the products such as 'price' or category information, we focus on extracting more information from the textual attributes, 'title' and 'short_description'. Our features fall into the 9 groups as explained in Table 2.

3.1.2 Feature Selection. For feature selection, based on estimators such as linear or tree-based models, one can identify the most importance features as well. For instance, linear models which penalized with the ℓ_1-norm have sparse solutions and most of their estimated coefficients are zero. Non-zero coefficient of a feature, therefore, shows its importance. In our work, we use linear SVM to identify our best features as listed in Table 3. We can see that

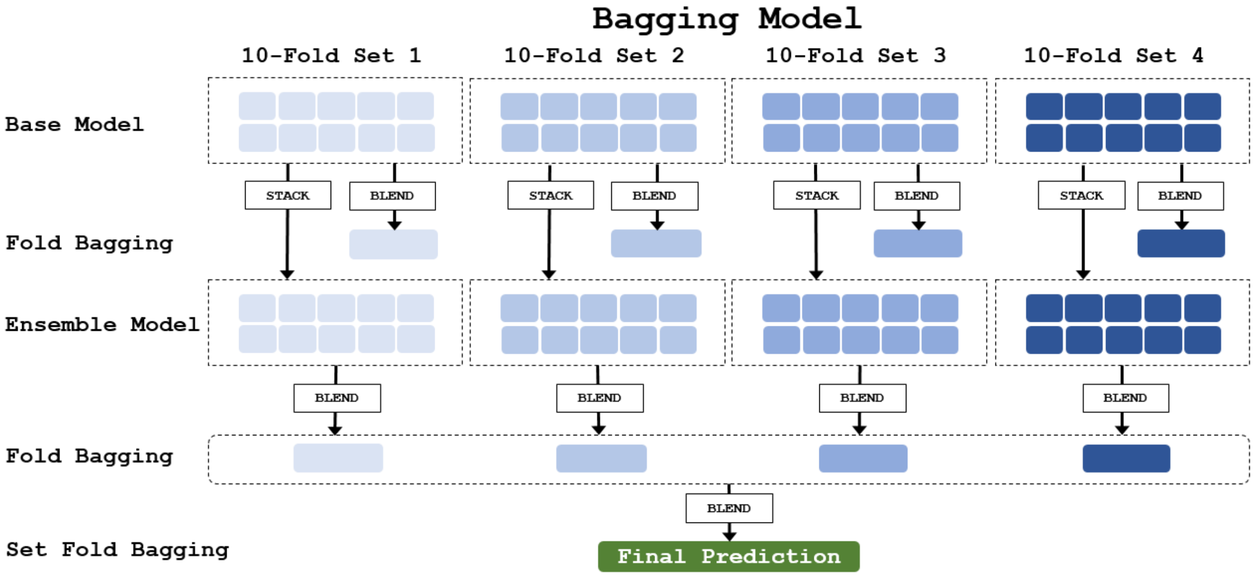

Figure 2: Product title quality modelling.

'sexy' and 'purse' are among the important features of clarity models, namely top_clarity. These features often come from Fashion category. On the other hand, 'my' and 'ph' are important features of conciseness models, namely top_conciseness. These feature represent the location where the products are being sold. We use these top features to train our predictive models which will be presented in the next section.

3.2 Product Title Quality Modelling

As shown in Table 1, there is disagreement in the training labels where the same titles have different labels. Therefore, if we use the whole training data set, these noises make machine learning model work worse. However, if one uses less training data, one may loose some important information. This leads to building less accurate models. To overcome the issue, we propose a bagging method which not only leverages all training data but also reduces noises in the training data. The proposed approach is shown in Figure 2 which consists of fold set generation, base model training, ensemble model building, and blending.

3.2.1 Fold Set Generation. To reduce label noise in the training data, we manage to use subsets of training data to build our models. In order to do that, we use 10-fold cross validation process to split the data into 10 separate folds. As the data is randomly partitioned, we don't know which one is the best split. Therefore, we apply the k-fold process 4 times to get 4 different fold sets. We then use 9 folds to train our machine learning models and make predictions for the testing data. So for each fold set, we have 10 models and 10 testing predictions, we finally blend 10 predictions to have the final predictions.

3.2.2 Base Model. We use stratified 10-fold cross validation process to evaluate the performance of our models. We train different estimators such as extreme trees (ETC), random forest (RFC), stochastic gradient descent (SGD), logistic regression (LOR), ridge regression (RDG), naive bayes (NBC) using Scikit-Learn [4]; extreme gradient boosting (XGB) [1]; light gradient boosting (LGB); and word2vec (W2V).

The performance results are shown in Figure 3 and Figure 4 for conciseness and clarity labels, respectively. In both labels, LGB has the RMSE of 0.3198 which outperforms other algorithms, XGB is the runner up and the next one is Logistic Regression. Finally, the worst model is Naive Bayes algorithm whose RMSE is 0.4047.

Similarly, LGB is also the best algorithm for predicting clarity label. While clarity RMSE of LGB is 0.2098, that of XGB is 0.2102. Notably, W2V model works quite well on clarity label. Its performance is just after logistic regression (LOR) and it also contribute to the performance of ensemble model that will be discuss in the next section. Although other algorithms such as naive bayes (NBC) and stochastic gradient descent (SGD) do not work well in this data, they can help improve the performance of ensemble models. Therefore, we keep these models in our final solution.

For each algorithm, we build 10 models and make 10 testing predictions. The final testing output is generated by blending these 10 predictions. We repeat these process on 4 fold sets as mentioned in the previous section.

3.2.3 Stacked Ensemble Model. In order to combine base models together, we apply stacked ensemble method. It means that we used validation predictions to train another models and make testing predictions using testing predictions of base models. We have tried a few algorithms for stacking but we found that XGB is the best. Therefore, we only use XGB for our ensemble models. The performance of ensemble models are shown in Table 4 where RMSEs of XGB for both conciseness and clarity are 0.31553 and 0.20745 respectively. Comparing to the best base model, the improvement is 0.004 for conciseness and 0.002 for clarity.

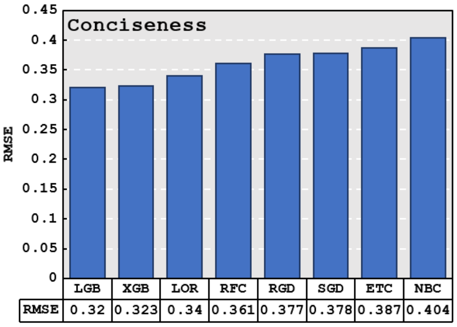

Figure 3: Base model performance for conciseness.

Table 4: Ensemble model performance.

label	algorithm	RMSE
conciseness	XGB	0.31553
clarity	XGB	0.20745

3.2.4 Model Selection. We also study the importance of base model in ensemble models. The results are shown in Table 5 and Table 6. Interestingly, while the most important models for conciseness are LGB models, those for clarity are SGD and W2V models. It means that clarity label is sensitive to noise so linear models can help reduce it significantly. Moreover, RDG trained on clarity label is also a good contributor of ensemble conciseness models.

3.2.5 Final Solution. Our final solution is the blended predictions of 4 fold sets. After training ensemble model and making predictions for each fold sets. We first combine 10 models for each fold set to have 4 testing predictions. We then blend the predictions by averaging them to have the final prediction.

4 LESSONS LEARNED

We have tried many NLP techniques such as stemming, stop-words removal, POS tagging, topic extracting but none of them worked. The key feature set for this problem is character ngrams. We found that term frequency works better than other unsupervised and supervised term weighting techniques [3]. Moreover, we tried dimension reduction technique such as latent semantic analysis (LSA) and principle component analysis (PCA) but they did not work. We also used recent deep learning techniques like attention model and we failed to have a useful model. Maybe short description is irrelevant to this problem or the QA staffs have not considered it when labelling the data.

5 CONCLUSION

In this paper, we have presented our winning solution for product title quality analysis. The proposed approach which uses powerful machine learning algorithms such as light gradient boosting (LGB), extreme gradient boosting (XGB) with traditional features in text mining such as word n-grams and character n-grams gives state-of-the-arts performance. We hope that our solution will probably provide a benchmark for solving product title quality analysis as well as other text classification problems in e-commerce.

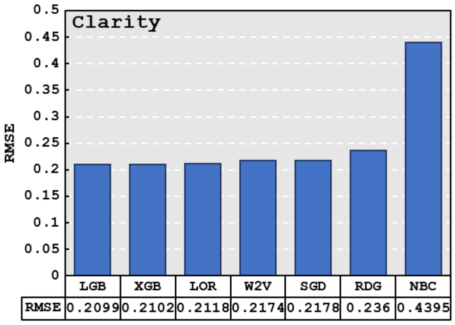

Figure 4: Base model performance for clarity.

Table 5: Top base models for conciseness.

algorithm	feature set	importance
SGD	title.boc.6grams	0.017230835
W2V	glove.twitter.27B.200d	0.015874704
NBC	title.boc.5grams	0.015768725
LOR	title.boc.6grams	0.014890845
LGB	title.boc.6grams, xg_feat title.color, title.brand, title.glove.twitter.27B.25d_mean price_feat, entropy_feat, top_clarity, char_set_feat	0.013428351

Table 6: Top base models for clarity.

algorithm	feature set	importance
LGB	title.boc.6grams, xg_feat title.color, title.brand, item_cnt, cat_cnt_feat, title_cat_feat	0.02956636
XGB	giba_clar	0.020367937
RDG	title.boc.6grams, sp_feat, title.color, title.brand	0.015111695
XGB	title.boc.6grams xg_feat title.color title.brand item_cnt cat_cnt_feat title_cat_feat	0.013797635
SGD	title.boc.5grams, sp_feat	0.011826544

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016.* 785–794. https://doi.org/10.1145/2939672.2939785

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

[3] Tam T. Nguyen, Kuiyu Chang, and Siu Cheung Hui. 2011. Supervised term weighting for sentiment analysis. In *2011 IEEE International Conference on Intelligence and Security Informatics, ISI 2011, Beijing, China, 10-12 July, 2011.* 89–94. https://doi.org/10.1109/ISI.2011.5984056

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.