CIKM AnalytiCup 2017 - Lazada Product Title Quality Challenge: A Bag of Features for Short Text Classification

Minh C. Phan Yi Tay
School of Computer Science and Engineering, Nanyang Technological University, Singapore
phan0050@e.ntu.edu.sg;ytay017@e.ntu.edu.sg

ABSTRACT

This technical report describes our solution in the CIKM AnalytiCup competition held in conjunction with the 26th International Conference on Information and Knowledge Management (CIKM 2017). The task at hand is to determine the quality of product tittles created by public users on the e-commerce platform Lazada. In this report, we present an insightful analysis on the dataset. The observations found are used to derive extensive set of features covering lexical, syntactic and semantic aspects of a title. Our solution for the problem is straightforward and effective. We use Gradient Boosting Tree with all the designed features. Furthermore, 2-level stacking is utilized to further improve the performance.

KEYWORDS

Text Classification, Text Mining

1 INTRODUCTION

Thousands and millions of products exist on e-commerce platforms often spanning across a myriad of product categories. From a shopper's perspective, products are mainly characterized by their name (or product title). As such, the naming of the product holds a majority stake in capturing the attention of the shopper. Consequently, sellers may resort to disruptive and toxic efforts to attract the attention of customers or game their search relevancy score. For example, coming up with titles such as *'hot sexy red clutch rug sack travel backpack unisex cheap with free gift'* may degenerate the overall user experience by cluttering the e-commerce platform with irrelevant, misleading titles.

Problem Definition This competition focuses on two main objectives, namely *Clarity* and *Conciseness* which are described as follows:

- **Clarity** - The ease of readability and delivery of key product attributes such as color, size, model, etc. Product titles with high clarity scores should be easy to understand and interpret quickly.
- **Conciseness** - The optimal point of conveyed information with respect to product title length, i.e., the amount of redundant content in the product title. Product titles that do not contain all necessary information also violate this quality.

Table 1: Number of samples in each dataset partitions and the ratio between positive samples (pos) and negative samples (neg).

Statistics	Training	Test$_1$	Test$_2$
Number of samples	36,283	11,838	12,674
Clarity (pos/neg)	0.943/0.057	_	_
Conciseness (pos/neg)	0.685/0.315	_	_

The problem is framed as two separate regression tasks which try to predict a score within $s \in [0, 1]$, each for clarity and conciseness. The labels provided, however, are binary (0 or 1).

This technical report is organized as follows. First we describe the product title dataset provided by Lazada[1] and three important observations made from the given dataset. Base on the observations, we introduce a set of features designed to capture special characteristics of the dataset and the problem. Lastly, we describe the use of Gradient Boosting Tree and 2-level stacking as our solution.

2 DATASET AND ANALYSIS

The dataset is provided by Lazada containing samples that are manually labelled by Lazada internal team. The dataset is split into three partitions detailed in Table 3. The largest split is the training data, containing more than 36 thousand product titles. The two other testing partitions contain more than 11 and 12 thousand product titles respectively. The labels for the two partitions are not revealed and they are used to update the leaderboard in the first and second phase of the competition [2].

Each product title in the dataset contains 9 attributes described in Table 2. In reprocessing step, we replace all the empty field with 'NA' and extract the text in the description (which is originally in HTML format) by using BeautifulSoup library [3]. To preserve the original information, we use the simple space-based tokenizer to split words in title and description. Next, we present some observations made from the training dataset.

Observation 1. *Unclear and non-concise titles usually contain duplication.*

Considering the following example in the training data *"Men Wallet Leather Cluth Bag Long Wallets Man Coin Purse Passport Holder Mens Credit Card Holder Men Purses"*, the words *'Men'*, *'Wallet'* and *'Holder'* appear more than once. Furthermore, there are

CIKM Cup"17 , Singapore, Singapore

978-x-xxxx-xxxx-x/YY/MM. . . $15.00
DOI: xx.xxxx/xxxxxxx.xxxxxxx

[1] A major e-commerce platform in Asia

[2] https://competitions.codalab.org/competitions/16652#learn_the_details-evaluation

[3] https://www.crummy.com/software/BeautifulSoup

Table 2: List of attributes of each sample in the dataset.

Name	Description
country	The country where the product is marketed, with three possible values
sku_id	Unique product id
title	Product title
category_lvl_1	General category that the product belongs to
category_lvl_2	Intermediate category that the product belongs to
category_lvl_3	Specific category that the product belongs to
short_description	Short description of the product, which may contain html formatting
price	Price in the local currency
product_type	It could have three possible values: 'local', 'international' or 'NA' (means not applicable).

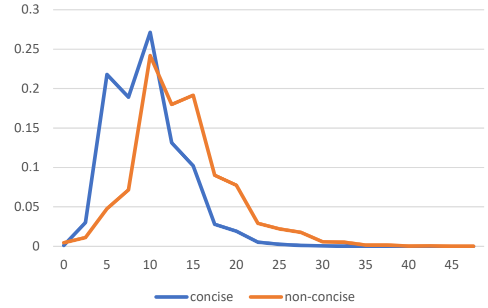

Figure 1: Title length distribution of concise and non-concise titles.

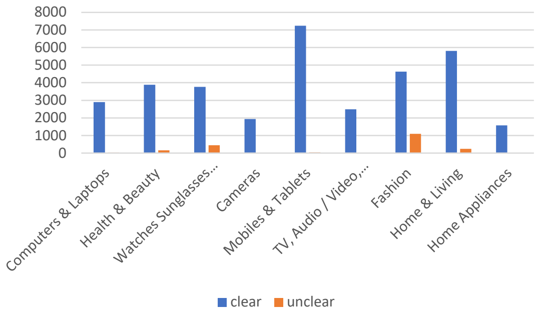

Figure 2: No of clear and unclear title in level-1 category.

words with similar semantic meaning such as *'bag'* and *'purse'*. The redundancy indicates that the title is likely unclear and not concise.

OBSERVATION 2. *Non-concise title is generally longer than concise title.*

Since non-concise titles may contain unnecessary or duplicate words, they are longer than the concise ones. To verify the hypothesis, we plot the distribution of numbers of words in positive and negative titles. As shown in Figure 1, the non-concise title tends to have more words than the concise one.

OBSERVATION 3. *The quality of a product title heavily depends on its category.*

We count the number of clear and unclear titles in each category. As shown in Figure 2, only category 'Fashion' and 'Watches' have high percentage of unclear title. For other categories, most of the titles are easy to read and understandable. Specially, for 'Home Appliances' and 'Cameras', only 0.4% and 0.5% of the titles are unclear. This observation hints that the categorical information is quite important for predicting the title's quality.

3 BAG OF FEATURES

In this section, we detailed 5 groups of features used to predict the quality (clarity and conciseness) of a product title. The first two groups of features capture multiple aspects of a product title. They are derived directly from the surface's form of title, description and the product's attributes. Furthermore, base on Observation 1, we aim to measure the similarity between words appearing in title. Therefore, we utilize Brown clustering and word embedding. Specifically, the words with similar semantic and syntactical meaning are placed in the same cluster and near each other in the embedding space. Finally, we use topic model to capture the latent category of product that is also useful for the prediction (see Observation 3).

3.1 Lexical and Categorical Features

These features are extracted based on the surface form of title and description. They cover multiple aspects of the product title.

Title and description length.

- Length of title intern of number of characters and number of words, *i.e.,* $len(title_w)$, $len(title_c)$
- Average length of word in title, *i.e.,* $len(title_c)/len(title_w)$
- Length of description intern of number of words, *i.e.,* $len(desc_w)$

Word duplication.

- Number of unique word in the title. $len(set(title_w))$
- Number of duplicate words in the title, *i.e.,* $len(\{w|w \in Set(title_w) \wedge freq(w) > 1\})$

Word popularity. Let $pop(w)$ is the frequency of word w in the whole training corpus, we derive features related to the popularity of all words appearing in the title and description:

- Min, max and average of $\{\log(pop(w)+1)|w \in title\}$
- Min, max and average of $\{\log(pop(w)+1)|w \in description\}$

Digits. Many title and description contains digits that indicates the product's model therefore, we count the number of digits appear in title and description as features. Furthermore, we also include the Jaccard similarity between the two set of numbers.

- $len(\{w|w \in title_w \wedge is\_numric(w)\})$
- $len(\{w|w \in desc_w \wedge is\_numric(w)\})$
- $Jaccard(Set(\{w|w \in title_w \wedge is\_numric(w)\}), Set(\{w|w \in desc_w \wedge is\_numric(w)\})$

Popular words. Table shows top 20 popular word and its frequency. Many of these words are good indicators that can tell us useful information about the product such as brand (*e.g.,* 'samsung'), model (*e.g.,* 'galaxy'), colour (*e.g.,* 'black'), fine-grained category (*e.g.,* 'iphone', 'case', 'cover'). Therefore, we create a frequent-word vocabulary that contains the top 1000 frequent words from the

Table 3: Top 20 popular words together with their frequency in training data.

Word	Frequency	Word	Frequency
for	10282	usb	2097
case	4965	black	2082
intl	4111	phone	2002
with	3833	samsung	1995
cover	3016	iphone	1842
women	2610	and	1788
leather	2553	new	1588
watch	2462	mini	1572
fashion	2410	galaxy	1562
(black)	2266	set	1456

training data. For each title, we include frequency counts of the top frequent words in the title as features.

- $\{freq\_count(w, title_w) | w \in 1000\_freq\_words\}$

Last word. Similarly, we collect the top 500 frequent words that appear as the last words in training titles (denotes as 500_last_words). We include features to indicate whether the last word in a title appearing in the 500_last_words.

- $\{title_w[-1] == w | w \in 500\_last\_words\}$

Upper-case.

- Whether the first letter in title is upper-case, *i.e., is_upper*$(title_c[0])$
- Whether the title is upper-case, *i.e., is_upper*$(title_c)$
- Whether the description is upper-case, *i.e., is_upper*$(desc_c)$

Matching between title and description.

- Length difference between title and description, *i.e.,* $len(desc_w) - len(title_w)$ and $len(desc_w)/len(title_w)$
- Number of common words, *i.e.,* $len(set(title_w) \cap set(desc_w))$
- Jaccard similarity between two sets of words in title and description, *i.e.,* $Jaccard(set(title_w), set(desc_w))$

Categorical features. We convert product category, country and product type attributes into one-hot representations and include them as features. Furthermore, we include the number of matching words between title and category name, in each of three levels:

- $len(Set(title_w) \cap Set(cat1_w))$
- $len(Set(title_w) \cap Set(cat2_w))$
- $len(Set(title_w) \cap Set(cat3_w))$

Price. We use the price in the original currency as well as the equivalent price in USD as two features, *i.e.,* $price$ and $price * to\_usd(price, country)$

3.2 Syntactic features.

Noun chunk. We use Spacy to extract noun chunks for a title.

- Number of noun chunks, *i.e.,* $len(nchunks)$
- Total length of all noun chunks in a title, *i.e.,* $\sum_{nc \in nchunks}(len(nc))$
- Number of words not in any noun chunk, *i.e.,* $len(\{w | w \in title_w \wedge w \notin nchunks\})$
- The lengths of three longest and shortest noun chunks.

Table 4: Example of words and their Bown cluster.

Cluster	Word	Cluster	Word
1100010	n1samsung	1110	purse(purple)
1100010	galaxynote2	1110	earrings(white)
1100010	p905	1110	earring(export)
1100010	galaxys6edge+	1110	hat(export)(intl)
1100010	galaxys4	1110	clothes-navy
1100010	galaxya3(multicolor)	1110	scarves(export)(intl)
1100010	galaxya9(multicolor)	1110	cover(multicolor)
1100010	4s/galaxy	1110	sunglasses(export)

POS tag. We use nltk and spacy to extract the pos tags of words in a title. After, we includes the frequency count of each POS as features, *i.e.,* $\{freq(pos) | pos \in ALL\_POS\}$.

3.3 Brown Clustering Features

A major challenge dealing with user-created data is the informal use abbreviations and vocabulary. To partially address this challenge, we adopt the Brown clustering algorithm, a hierarchical clustering algorithm which groups the words with similar meaning and syntactical function together [2]. The intuition of the algorithm is that similar words have similar contexts in which they occur. The clustering is then conducted by maximizing the mutual information of the bi-gram language model [1].

Given all titles in the training data, by applying the Brown clustering algorithm, we obtain a collection of clusters. Each word belongs to exactly one word cluster. Reported in Table 4, words that indicate the model of product (*e.g.,* 'galaxynote2', 'galaxys4') or product type (*e.g.,* 'purse(purple)', 'earrings(white)') are grouped into the same cluster.

To capture multiple aspects of semantic and syntactic clustering, we utilize multiple clustering outcomes. Specifically, we run Brown clustering algorithm on the text corpus with 4 different number of cluster settings: 60, 80, 100 and 120 clusters. For each clustering setting, we count the number of title words fall into each of cluster as feature, expressed as follows:

$$\{len(\{w | w \in title \cup w \in c\}) | c \in brown\_clusters\}$$

3.4 Word Embedding Features

To capture the soft-matching (*e.g.,* 'case' vs 'cover'), we employ the word embeddings. The similarities between word embeddings in a title can be a good indicator for evaluating the duplication,*i.e.,* high pair-wise similarities indicate that the words in title are highly overlapped in meaning therefore the title may not be concise.

In this work, we use ConceptNet Numberbatch pre-trained embedding [6]. Furthermore, we also create a domain-specific word embedding from the provided data. Specifically, we collect all the titles and description sentences from the given data as a text corpus. We filter out tokens whose frequency are less than 3. Finally we use Gensim [5] to train skip-gram [4] embeddings with dimension of 100, window size is 5. For each of the embedding scheme, we calculate the pairwise similarities between every pair of words in a title. We include the top 2 highest and lowest similarities as features.

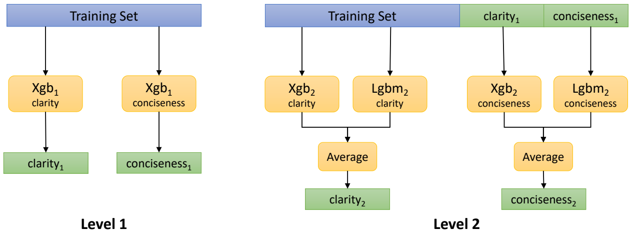

Figure 3: Two-level stacking architecture.

3.5 Topic Model Features

We use Latent Dirichlet Allocation (LDA) [3] to extract the topics presenting in each title. The number of topics is set to 50 and the topical distribution of a title is used as features.

4 EXPERIMENT

Evaluation metric. Submissions will be evaluated using the Root-Mean-Square Error (RMSE), defined as follows:

$$RMSE = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}} \tag{1}$$

In the above formula, N is the number of instances, y_{pred} is the predicted probability value for a given instance, and y_{ref} is the ground-truth value for that instance.

Experiment setting and model selection. Although the evaluation metric is regression-typed, we treat the problem as classification task and use the raw probability outputted from a binary classifier as the prediction. We tried different models including neural net work specifically Multilayer Perceptron, Random Forest, Extra Trees and Gradient Boosting. Among all of them, Gradient Boosting yields the best performance for both clarity and conciseness. Therefore, we decided to use Gradient Boosting, with two implementations, namely, xgboost [4] and lightgbm [5]. It is worth mentioning that the performance by xgboost is slightly better than lightgbm however lightgbm is much faster.

We use a 2-level stacking ensemble illustrated in Figure 3. In the ensemble, the clarity and conciseness prediction by xgboost will be used as extra features for training the second-level xgboost and lightgbm. Finally, the clarity and conciseness predictions are the average of the outputs from the second-level xgboost and lightgbm models.

We use 10-fold cross validation to tune the model hyper parameters. Details of the settings can be found in the source code. All the experiments are run on a Xeon processor workstation with 40 threads. The whole training and predicting process takes about 6 hours.

Experiment results. We report the performance of level-1 xgboost model with 10-fold cross validation on the training dataset. We also do an ablation study in which we disable one group of features and report the performance with the ablation. As illustrated in Table 5, decent performance is achieved by using the only the basic lexical and categorical features. Adding Brown clustering and word embeddings features further improves the performance by 0.002 and 0.004 RMSE for clarity and conciseness respectively. We believe that if there are more training data, the two groups of feature will be more useful. Finally, we report the leaderboard performance (Phase 2) in Table 6. By using stacking and average ensembles, we archive about 0.002 reduction in RMSE for both clarity and conciseness predictions.

Table 5: Clarity and conciseness performance (in RMSE) of level-1 Xgboost with 10-fold cross validation. The best results are in boldface and the second-best are underlined.

Setting	Clarity	Conciseness
Baseline (class ratio as prediction)	0.231150	0.464382
Xgboost$_1$ (lexical and categorical)	0.209935	0.321637
Xgboost$_1$ (all features)	0.207932	0.317449
No syntactic features	**0.207795**	0.318053
No Brown clustering	0.208507	0.318081
No word embeddings	0.208219	0.318934
No LDA	0.208093	**0.317385**

Table 6: Leaderboard performance (Phase 2).

Setting	Clarity	Conciseness
Xgboost (level 1)	0.244867	0.333155
Average ensemble (level 2)	0.242820	0.331480

5 CONCLUSION

In this paper, we present a straightforward and effective solution for predicting the quality of a product title. Although we have tried some deep learning models such as convolution neural network and recurrent neural network, none of them shows comparable performance to Gradient Boosting method. The possible reason could be there is not enough training data. On the other hand, Gradient Boosting seems to be the most effective supervised approach in this case. Furthermore, we believe that the extensive set of features we introduced is applicable for not only this task but also other text mining problems.

REFERENCES

[1] John Blitzer and Xiaojin Zhu. 2008. Semi-Supervised Learning for Natural Language Processing. In *ACL*. 3.

[2] Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics* 18, 4 (1992), 467–479.

[3] Matthew D. Hoffman, David M. Blei, and Francis R. Bach. 2010. Online Learning for Latent Dirichlet Allocation. In *NIPS*. 856–864.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[5] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *LREC*. ELRA, 45–50.

[6] Robert Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. (2017).

[4] http://xgboost.readthedocs.io/en/latest

[5] https://github.com/Microsoft/LightGBM