CIKM AnalytiCup 2017 – Lazada Product Title Quality Challenge

Multi-Deep-Ensemble For Product Title Quality Scoring

Hong Diao WEN
University of Electronic Science and Technology
Cheng Du, Si Chuan
uestc_wen@outlook.com

ABSTRACT

Although NLP techniques have been employed for many applications,product title quality scoring remains a tough problem for the lack of semantic words.In this paper,we present a multiple deep ensemble method for the challenge. In our experiment, feature engineering and selection is essential for all of models,while the effect of some features are intuitive or relative to exact task.After proper feature preparing,we projected tree-based models like xgboost,lightgbm,random forest and adaboost as our start,then we used up to 7 different models to perform ensembling,we concatenated category embedding features and numeric features with the outputs of recurrent neural network that derived from word embedding to cover the sentiment meanning in addition,we weighted 4 models that was based on previous ensemble models together as our output at last.We won first place on conciseness task of CIKM AnalytiCup 2017: Lazada Product Title Quality Challenge,which indicates that our method is effective.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Machine learning** → **Ensemble methods**; • **Artificial intelligence** → **Natural language processing**;

KEYWORDS

Feature engineering, ensemble methods, semantic analysis

ACM Reference Format:
Hong Diao WEN. 2017. CIKM AnalytiCup 2017 – Lazada Product Title Quality Challenge. In *Proceedings of ACM CIKM conference, Singapore, October 2017 (CIKM'17),* 4 pages.
https://doi.org/10.475/123_4

CIKM'17, October 2017, Singapore

ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

1 INTRODUCTION

Sentiment analysis[1] is a basic and hot topic in natural langauge processing.Towards this challenge,many nice models like recurrent neural network (RNN)[2] have been proposed, RNN performs well for its convenience to capture sequence evaluation between words or terms,where the word is represented by vector which is embedded from id in the dictionary.Despite its powerful ability,RNN is hard to sense statistic implicit.In this paper,we proposed a method to combine statistic features with semantic features together using classifiers to solve this problem.

Feature engineering plays an import role in most of machine learning models,even deep learning prefer inputs that are distinguished though its powerful ability of non-linear feature extraction.In this challenge,we pay much attention to prepare features.To start with,as manaul factored features quantifing documents easily,we presented with sufficient numeric features basicly,we found some of them are interesting and would like to specify it later.Then,the bag of words methods like word count have been proven to be efficent in some kinds of application,we adopted this as a part of our method, then we applied matrix decompostion method like SVD[3] above it to perform Linear Discriminant Analysis(LDA)[4].

Ensemble methods[5] are common to boost performance for different classifiers are non-linear even fed with same features.To ensemble various of features and models,we used task stacking and model stacking approaches to generate meta featrues.In addition,our deep learning model were built on the trained Glove word embedding[6] and numeric features including meta features,we also implemented category embedding[7] which is similar with word embedding,the method is nice for the capability to combine semantic intent with statistic meaning.At last,we just weighted multiple models as our final output.

Our solution is organized as follows:first we summarize our process of feature preaparing,then we conclude the ensemble methods we adopted,next we specify our deep learning method to manage all the features,we give a conclusion at last.

2 RELATED WORK

As the popular usage of deep learning,nlp challenges have been improved much.However,the data is not enough to train a deep model when the words of bag are mainly entities and samples are less.There have been many attempts to

combine deep learning models with traditional statistical models together in nlp.Here we propose a method to ensemble LSTM[8] with meta features generated by popular classifiers in machine learning,which works well in our experiments.

3 FEATURES AND MODELS

3.1 Feature Preparing

In this section,we would list our features and explain our insights to some of them.

3.1.1 Statistic Features. In our statistic feature engineering,we covered features like number of word, number of char,number of entity,number of error typed word, number of duplicate word,number of upper char for each word,number of occurence of each word,number of intersection words between title and description,LDA on TF,category frequency,word embedding,category embedding and so on.We would specify some features as they are intereting and effictive.

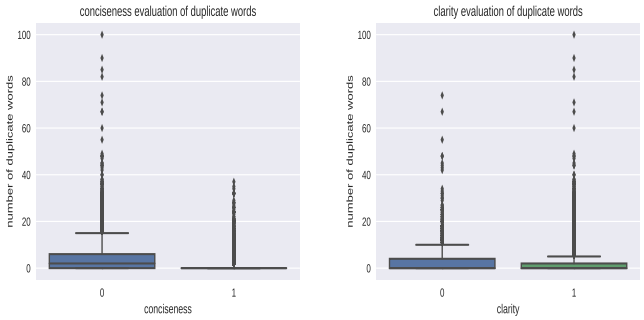

Figure 1: The illustration of different distributions of duplicate words number between conciseness and clarity.This indicates that we should select duplicate words number as a feature for conciseness but not clarity.

To start with,we used box plot as showed in Figure 1 to illustrate that the number of duplicate word is a good feature for the task of conciseness but not clarity. Found the insight of task defference,we always tried to implement feature selection throghout our models.

Then we constructed TF(term frequency) of title,where we limit number of word counter as 4000 and 2800 for conciseness and clarity separately.We projected TF rather than TF-IDF(term frequence inverse document frequency) just because the former performs better.We also used TF of conciseness to implement Linear Discriminant Analysis(LDA) towards category level 3.

Formally,let ω be the indication of line where the data maps to,we can derive the representation of ω as follows:

$$S_b W = \lambda S_\omega W \tag{1}$$

where within-class scatter matrix S_ω defines as the scatter matrix sum of every class:

$$S_\omega = \sum_{i=1}^{N} S_{\omega_i} \tag{2}$$

$$S_{\omega_i} = \sum_{x \in X_i} (x - \mu_i)(x - \mu_i^T) \tag{3}$$

and between-class scatter matrix is in the following form:

$$S_b = \sum_{i=1}^{N} m_i (\mu_i - \mu)(\mu_i - \mu)^T \tag{4}$$

μ and μ_i represents mean vector of whole data and class i respectively,W has closed form of solution which is composed of N-1 eigen vector corresponding to the N-1 biggest eigen value of $S_\omega^{-1} S_b$,while S_ω^{-1} can be solved by SVD:

$$S_\omega = U \Sigma V^T \tag{5}$$

where Σ is a diagnal matrix ordering by singular values of S_ω,generally we can select k largest singular value as main factors which can reduce the impact of noise at the same time.

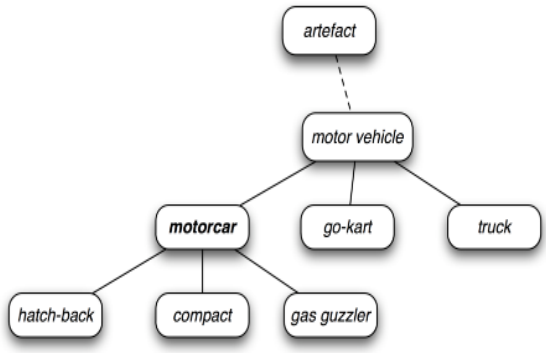

Figure 2: A sample hierarchy tree of WordNet

In addition to these features,we also employed some useful tools like NTLK[9] WordNet,Spacy english word document.Spacy was used to create number of word in english dictionary.WordNet synsets correspond to abstract concepts, and they don't always have corresponding words in English. These concepts are linked together in a hierarchy. Some concepts are very general, such as Entity, State, Event these are called unique beginners or root synsets. Others, such as gas guzzler and hatchback, are much more specific. A small portion of a concept hierarchy is illustrated in Figure 2.We created title depth and title path similarity based on WordNet.

3.1.2 Task Stacking Features. Though different tasks indicate different insights,we recognized that the labels of clarity and conciseness are related where the clarity 0 is absolutely crresponding to conciseness 0,then we used stacking method to capture prior information from each other.We performed k-fold cross validtion on training dataset,in other words,we

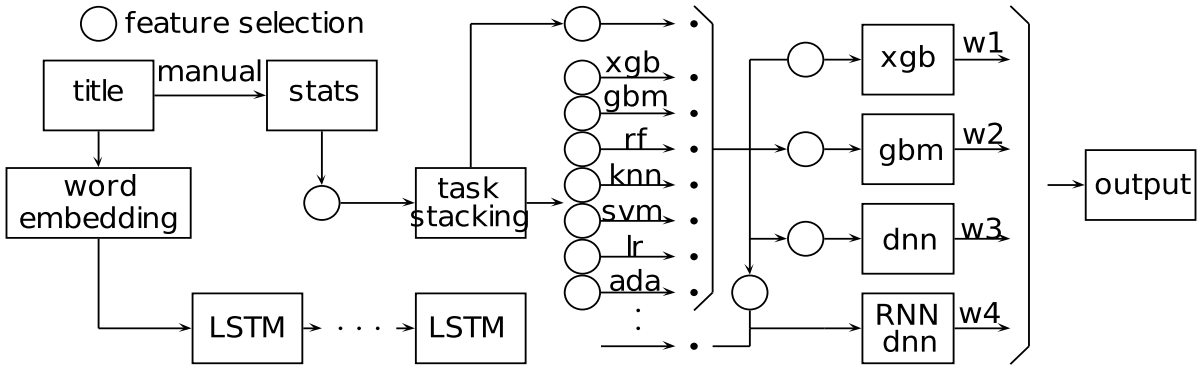

Figure 3: The overall view of our model.The upper channel first goes to form manual-factored features and employ task stacking,then ensemble with several classifiers using stacking method too.The down channel feeds with word embedding vector and recursively extract hidden state at each time.Then the deep ensemble combines two channel s together with 4 different models and weighting them all to generate final result.

trained on k-1 folds of samples in training set then predict on the rest fold of samples and all the samples in test set at every iterate, meanwhile we trained k models with same parameter setting but different subfolders.The stacking scheme can be viewed in Figure 4.

In order to cover the task correlation prior better,we choosed several tree-based models for it is resilience with missing values and category inputs,we used xgboost[10],lightgbm,random forest and adaboost built upon decision trees in our experiment.After this period,we construct new dimensions of task features based on different models.The procedure of task stacking in our whole model can be found in Figure 3.

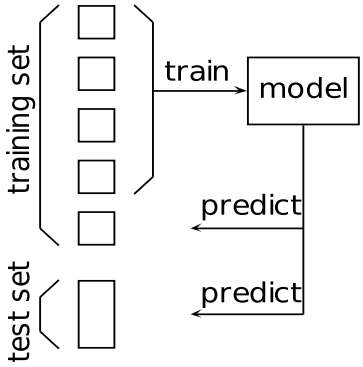

Figure 4: Stacking scheme in both task and model stacking.Note that different models trained with diffrent k-fold data split seeds.

3.2 Multi-Deep Ensemble

3.2.1 Model Stacking. After the stacking of tasks,we have the generated stacking features and numeric features from previous stage.Enjoying the selection of numeric features,we can further perform ensembling by the means stacking almost without limitation.

We have took advantage of more than 7 traditional classifiers,including xgboost,lightgbm,random forest,knn,linear svc,linear regression,adaboost,mlp and naive bayes to generate a more robust ensemble model with different data split seed respectively.To ensemble more non-linear models,we tried to enroll same kind of xgboost with different parameters and object functions.

After the first stage of model stacking,the meta features generated by models are all continious and comparable,but they may make confuse of models within a small range.Here, we introduce a corse-to-fine scheme to better mining the distribution of meta features.

Given a continious feature x_i,split the feature into m ordered parts and indicate each part by x_i^m based on values:

$$x_i = \sum x_i^m \tag{6}$$

where $x_i^{m-1} < x_i^m$,then a new feature can be created as:

$$x_j = m \tag{7}$$

we split feature from xgboost into 10 parts and performed one-hot encoding in our experiments.

Then we applied deep ensemble based on all the features from previous stage.we took xgboost,lightgbm and dnn as three components of output models.

3.2.2 Combined LSTM. The deep recursive model is intend to mining the semantic meaning in the product title.First,we performed word to vector through trained Glove embedding.Then the title can be projected as a sequence prediction problem where the deep recursive neural network fits well.However due to numerous entity words in this problem,the correlation bettween words of title is less, we proposed to ensemble numeric features from statistic features and meta features including category embedding with RNN based on LSTM block.The detail architecture can be viewed in Figure 3.The category embedding can be viewed as a model which maps category ids to embedding vectors using fully connected network,it learns to quantilize the distance and relationship

Table 1: Performance Of Models

Models	RMSE
xgboost	0.3130
lightgbm	0.3141
dnn	0.3138
LSTM	0.3115

between categories and facilites the numeric modeling of RNN.

The RNN part tooks the embedding of word at each time and output with hidden state,the final output is the map of hidden state of all the times which is a vector.To built a more stong and robust model,we construct a two layer RNN model with 64 and 32 units respectively.Then we feed model with features from selected previous stage and the output vector of RNN to dnn in addition,where the dnn is setted as 3 layers with 64,128,192 neurons respectively and 0.1 portion of dropout at last layer,the effect of dnn is to combine statistic implicit and semantic meaning together.The LSTM model was trained end-to-end for prevous feature was prepared and we took it as another component of output models.

At last,we combined four models as xgboost,lightgbm,dnn and LSTM together by weighting.w1,w2,w3 and w4 were setted with 0.35,0.25,0.1 and 0.3 for conciseness,0.6,0.2,0.2 and 0.0 for clarity as which performed best in validation set.

3.3 Model Settings

Cause to the parsing of description and SVD method, it requires about 30min to finish feature preparing.We didn't tune classifier settings too much except for xgboost,lightgbm and RNN with LSTM,it may takes more than 1 hour to perform model stacking for the employ of random forest and adaboost.The LSTM needs Glove model which is about 8GB and less than 20min to run.We setted 4 threads as default if the model is possible to run in parrel.

4 LESSONS LEARNED

In our experiments,we found that bag-of-words feature like TF gives largest gain,this may lays in that words in this dataset is mostly entity.From this point,we believe the tagging of exact product entities in title mannaully or though trained tagging model would be effective,however we didn't perform this for time and ennergy consuming.Besides,the number of duplicate words is helpful for conciseness task as we wished.In addition, the number of upper char for each word is useful for detecting entity words as we tested,where the word with multiple upper chars are easy to draw attention but less likely to be readble.

5 ANALYSIS

The performance of different models in last stacking stage were showed in Table 1.The evaluation of xgboost and lightgbm were based on cross validation while dnn were based on leave out method where the ratio of validation set is 0.13.Overall,we reached a RMSE of 0.3284 in test set which ranking first in conciseness task.

6 CONCLUSIONS

In this abstract,we proposed a multi-deep-enseble method to solve the problem of product title quality scoring for the CIKM AnalytiCup 2017 – Lazada Product Title Quality Challenge.We combined traditional classifiers with deep recurrent model together,which is good for covering both statistic prior and semantic meaning.The public leaderboard shows that our method performs effective.We also believe that more concise tagging of entities in titles would be a promising way in the future.

REFERENCES

[1] Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. "An introduction to latent semantic analysis." Discourse processes 25.2-3 (1998): 259-284.
[2] Mikolov, Tomas, et al. "Recurrent neural network based language model." Interspeech. Vol. 2. 2010.
[3] Klema, Virginia, and Alan Laub. "The singular value decomposition: Its computation and some applications." IEEE Transactions on automatic control 25.2 (1980): 164-176.
[4]
[5] Wang, Gang, et al. "A comparative assessment of ensemble learning for credit scoring." Expert systems with applications 38.1 (2011): 223-230.
[6] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
[7] Cheng Guo, Felix Berkhahn. "Entity Embeddings of Categorical Variables." arXiv:1604.06737.
[8] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[9] Bird, Steven. "NLTK: the natural language toolkit." Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics, 2006.
[10] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.