Detection of Author’s Educational Level and Age based on Comments Analysis

A. Gomzin[1,2]
gomzin@ispras.ru

A. Laguta[1]
laguta@ispras.ru

V. Stroev[1,2]
stroev@ispras.ru

D. Turdakov[1,2,3]
turdakov@ispras.ru

[1] Ivannikov Institute for System Programming of the RAS
[2] Lomonosov Moscow State University
[3] National Research University Higher School of Economics

Abstract

The paper presents evaluation of machine learning based methods for prediction of author’s educational level and age using text she wrote in on-line social networks. The search for the best methods was carried out by a group of students who took an educational course in text processing. The paper describes the design of a practical task offered to students, developed in such a way as to motivate students to search for the best solutions. Evaluation results showed that the support vector machine with a linear kernel performs unexpectedly well in the given constraints.

Определение уровня образования и возраста автора на основе анализа его комментариев

А. Гомзин[1,2] gomzin@ispras.ru

А. Лагута[1] laguta@ispras.ru

В. Строев[1,2] stroev@ispras.ru

Д. Турдаков[1,2,3] turdakov@ispras.ru

[1] Институт системного программирования им. В.П. Иванникова РАН

[2] Московский государственный университет имени М.В. Ломоносова

[3] Национальный исследовательский университет Высшая школа экономики

Аннотация

В статье представлена экспериментальная оценка методов определения уровня образования и возраста автора на основе текста, который он писал в онлайновых социальных сетях. Поиск наилучших методов осуществлялся группой студентов, проходивших курс по обработки текстов. В статье описывается дизайн практического задания, предложенного студентам, разработанный таким образом, чтобы мотивировать студентов к поиску наилучших решений. В заданных ограничениях неожиданно хорошо показал себя метод опорных векторов с линейным ядром.

1 Introduction

Results presented in this work were obtained during practical part of text processing course[1] (TPC) for students of MSU and HSE. Our main goal was to give students the opportunity for solving open problems in NLP. Prediction of education level and age for comment's authors were selected as such open problem for the course in the fall 2017. We have developed practical session

[1] http://tpc.at.ispras.ru

to motivate students to use various machine learning techniques and to find the best solution.

On-line social networks allow users to fill their profiles that may contain demographic attributes values, such as age and education level. Profiles are not fully filled, so the task of unknown attributes detection arises. Explicit and predicted values are used in recommender and marketing systems. In particular, the predicted values can be used for inferring the target audience of marketing campaigns in the Internet.

Pennebaker and King [7] reported a correlation between the demographic characteristics of people and the stylistic features of their writing texts. Based on this observation we suggested a hypothesis that unknown education level and age of an author may be revived using public text content generated by her. The problem posed to course participants is to predict education level and age of on-line social network's users using their textual comments.

The structure of the paper is following. First, we survey articles related to demographic attributes inference in "Related work" section. Next, formal task definition, evaluation system setup and information about submitted solutions are given. Then evaluation results are presented. The conclusion describes the main findings.

2 Related work

The most interesting data sources for analyzing demographic attributes are social networks, such as Facebook, Twitter and others. Some studies analyze comments on Youtube [4], news and e-mails corpuses[3] and blogs.

The most common approach used for age, education level and other demographic attributes prediction is to extract features from users' texts and apply machine learning methods to them.

Age detection task was researched in [2]. Authors have analyzed 10K blog posts and found features that may be used for age detection: country, time of message publishing, message length, punctuation characters rate, subsequences (n-grams) of characters and words, interests, hyperlinks, pictures, etc.

Authors of [9] have predicted gender and age intervals using Multi-Class Real Winnow machine learning algorithm. The following features were used: words' frequencies for each gender and age interval value, unigrams, part of speech, functional words. Authors analyzed texts from 71000 blogs and have reached 70-80% accuracy.

Some studies aimed to predict not the age interval, but exact age value [5]. Age prediction task is reduced to regression task rather than classification task. Authors use n-grams and part of speech tags as features and linear regression as machine learning algorithm.

Age interval prediction task is considered in [6]. Authors analyze users that write in Dutch language. Character and word n-grams ($n = [1..3]$) are used as features.

Work [8] is derived to demographic attributes prediction in Twitter. Gender, age interval (≤ 30, > 30), political views, region are considered. Authors extract two type of features: socio-linguistic and n-grams. Socio-linguistic features are emoticons, abbreviations, repeated punctuation and other characters. SVM classifier is used for each feature set. Then, authors employed a stacked model where another SVM for this task is utilized. Stacked classifier features are the predictions from the n-gram and socio-linguistic models along with their prediction weights.

3 Age and education level detection in TPC-2017

This section describes details of practical session and evaluation results. First we describe the datasets. Next, formal task definition and restrictions for solution are presented. Finally, we observe solutions proposed by participants and present evaluation results.

3.1 Dataset

Experimental data is collected from public groups of Russian social network Vkonakte. We select one million most active public pages for data crawling. Then we group comments from these pages by author.

So, the corpus consists of users. Each user is represented as anonymized profile and set of public comments. The dataset is a random sample of users from all collected data.

Only users with at least 20 comments are included in the dataset. The language of comments is Russian.

Profile consists of two attributes: age and education level. Age is divided into the following intervals: “≤ 17”, “$18 - 24$”, “$25 - 34$”, “$35 - 44$”, “≥ 45”. Education level takes three values: “lower”, “middle”, “high”. “High” value means that the user have graduated an university; “lower” means that the user is currently studying at school; “middle” value means that the user have finished school and does not have higher education yet.

The corpus contains users with at least one of these attributes presented in her profile.

Users with $age > 18$ and "lower" education level in the profile are not included in the dataset. The same filter is applied to the users with $age < 15$ and education level "high" or "middle". We consider these combinations of age and education level unlikely.

After dataset collection we divide the data into three parts that are not intersected: Train, $Test_1$ and $Test_2$. Train part consists of 8607 users, $Test_1$ consists of 1070 users, $Test_2$ consists of 1053 users. Train data is shared with participants and used for classifiers training. $Test_1$ and $Test_2$ are used for evaluation. $Test_1$ is used for weekly evaluation, $Test_1$ and $Test_2$ are used for final evaluation.

3.2 Task definition and restrictions

Participants were asked to find solution for two following tasks:

- Age detection from the text. The input is a list of texts belonging to one author. The output of the algorithm should be one of the classes: "≤ 17", "$18 - 24$", "$25 - 34$", "$35 - 44$", "≥ 45";

- Educational level detection from the text. The input is the same as in the previous task. The output should be one of the classes: "lower", "middle", "high".

We also implemented two baselines. First one returns label of main class: "25-34" for the first task and "high" for another one. The second baseline is Linear SVM with word unigrams weighted by TF-IDF as features. Last baseline performs surprisingly well and outperforms other classifiers with same features and even with many other feature sets. A participant gets one score point if her solution is better then lower baseline and two score points if result is better than higher baseline.

In order to motivate participants for improving their solution even if they obtain good result we rank all solutions and grant additional score points for top-8 if they are better than baselines. First one get 8 additional score points, second one – 7 etc. Each participant could see her rank and improve it. These scores were taken into account in the issue of the final grades.

In contrast to Kaggle we run all solutions on our hardware in the restricted environment. This allows us to keep test set unseen by participants and put everyone to relatively similar conditions. Automatic testing system has the following limitation. It supports Python 3.x programming language. Commonly used python libraries (NLTK, scikit-learn, pythorch, keras etc.) are available. Package with solution can't exceed 15 Mb, maximum testing time for solution is 20 minutes, RAM is restricted to 16 Gb. Participants could submit pre-trained models. Each participant has 10 attempts in one week. Total duration of practical session — 9 weeks. Due to restriction for number of attempts, participants used cross-validation on training dataset in order to find well performing configuration and only then submit solution into the system.

In addition we train fasttext [1] models with 100, 200 and 300 dimensions on 3.3GB of user's posts and comments and make them available for download [2] as well as for the testing system.

3.3 Proposed solutions

№	Classifier	Range	Type	Tokenizer	Spec. features
1	Linear SVM	1-4	char_wb	default	C=2.5
2	Linear SVM	1	word	NLTK	sublinear_tf max_df=0.95 C=1 Stemming
3	Linear SVM	1	word	NLTK	sublinear_tf C=1
4	LSTM	1	word	default	max_feat=10^5
5	Linear SVM	2-4	char	default	ë → e sublinear_tf C=1
6	Linear SVM	1	word	TweetTokenizer	sublinear_tf C=1
7	Voting	3-4	char	TweetTokenizer	Stop words LOG C=148 SVM C=0.68
8	Linear SVM	1-4	word	TweetTokenizer	max_feat=$5 * 10^5$ C=15
9	Voting	3	char	TweetTokenizer	LOG C=75, SVM C=0.8
		1	word		
10	Log. Regression	3	char		C=17 word max_feat=$2 * 10^5$ char max_feat=$2 * 10^5$ Stemming
		1	word	default	

Table 1: Top 10 methods description.

[2] http://tpc.at.ispras.ru/dopolnitelnye-materialy/

Participants submitted 209 runs for age classification and 179 runs for education level classification. Submissions used linear SVM, logistic regression, naive Bayes, Passive-Aggressive classifier, SGD classifier, Ridge classifier and neural networks for classification. Word and char n-grams with TF-IDF vectorization or with just count vectorization and fasttext word embeddings were used as features.

We present details for 10 most accurate solutions in the Table 1. These solutions are based either on *Linear SVM*, *LSTM* or *Voting* classifier. Voting classifier used Logistic Regression and Linear SVM as estimators and choose final labels by selecting one with maximum probability. C is a regularization parameter for Logistic Regression and Linear SVM .

Range and *Type* columns contain N-gram range and N-gram type respectively. *Char_ wb* type stands for character n-grams created only from text inside word boundaries. *Sublinear_ tf* implies a solution uses sublinear tf scaling, i.e. replace tf with $1 + log(tf)$ while computing tf-idf weights. *Max_ df* is a threshold that allows to ignore terms with a document frequency strictly higher than it. $\ddot{e} \rightarrow e$ means that all «ё» letters are replaced with «е». *Default tokenizer* indicates that default tokenization algorithm from scikit-learn library was used. In addition two classifiers process words with Porter *stemmer*. *TweetTokenizer* is special tokenizer for twitter posts from NLTK library. *NLTK tokenizer* means that dafault tokenization algorithm from NLTK library was used.

All presented solutions used n-grams with TF-IDF vectorization for feature extraction. It extracts occurrence of a n-gram in a given document normalized with TF-IDF. This allows to extract more information by using frequencies of occurrence of a token in text corpus.

Surprisingly solutions that used fasttext vectors showed weak results. Thus they are not presented in the Table 1. Also there were some attempts to use n-grams with feature selection, but they also demonstrate low accuracy.

3.4 Results

For evaluation we use accuracy score measured on $Test_1$ and $Test_2$ datasets.

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i),$$

where $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, $n_{samples}$ is the size of dataset. Participants could evaluate their solution on $Test_1$ during development. $Test_2$ used only for final evaluation.

Results of described methods are shown in the Table 2. The table contains solutions and baselines (columns “Sol.”) and corresponding accuracy values (columns “Acc.”) of age and education attribute prediction sorted by $Test_1$ and $Test_2$.

Age				Education			
$Test_1$		$Test_2$		$Test_1$		$Test_2$	
Sol.	Acc.	Sol.	Acc.	Sol.	Acc.	Sol.	Acc.
1	0.61460	8	0.59656	5	0.7712	9	0.76671
2	0.61204	3	0.58995	1	0.76485	7	0.76419
3	0.61075	5	0.58995	6	0.76485	5	0.76167
8	0.60819	2	0.58862	7	0.75979	10	0.76167
4	0.60179	1	0.56614	9	0.75095	6	0.75284
5	0.60179	4	0.55423	10	0.74716	1	0.7465
bl_2	0.58259	bl_2	0.56481	bl_2	0.73957	bl_2	0.73140
bl_1	0.31626	bl_1	0.30556	bl_1	0.50569	bl_1	0.47793

Table 2: Results of age and level of education classification by comment texts (sorted by $Test_1$ and $Test_2$). bl_1 and bl_2 correspond to baselines.

Best results were achieved by solutions based on linear SVM. Also one neural network solution, namely LSTM, got into top-10. Note LSTM shows good results only on $Test_1$ for age detection task. However on the $Test_2$ its accuracy is even lower than baseline. All other submission that uses neural network algorithms demonstrated bad performance.

Three of ten methods (2, 3, 8) are in top-4 best solutions for both $Test_1$ and $Test_2$ for age classification; 2 of 10 methods (5, 7) are in top-4 best solutions for both $Test_1$ and $Test_2$ for education classification.

Five of ten best solutions are using character n-grams and two of them are in top four methods for both attributes. All top-3 solutions for both attributes are using linear SVM classifier.

4 Conclusion

In this paper we present an evaluation of machine learning based methods for prediction of author’s educational level and age using text she wrote in on-line social networks.

The search for the best methods was carried out by a group of students who took a course in text processing. They applied several methods for these tasks: linear SVM, logistic regression, naive Bayes, Passive-Aggressive classifier, SGD classifier, Ridge classifier and neural networks. They used word and character n-grams with TF-IDF vectorization or with just count vectorization and fasttext word embeddings as features. Best results were achieved by solutions using linear SVM. Surprisingly most neural network solutions demonstrated bad performance.

We can see that the results exceed baseline 2 insignificantly. This cause because baseline 2 applies the same classifier. Actually most of solutions are based on the SVM classifier and TF-IDF feature extractor parameters tuning.

We have two hypothesizes why neural networks showed low accuracy. First one is the restricted computational resources available for each solution. Another reason could be the instability of algorithms to noisy data. On-line social network users may provide wrong data in their profiles. We made only simple filtering of datasets using correspondence between the values of two related attributes, but it obviously is not enough. We plan to investigate influence of wrong profile attributes' values to different classifiers in the future work.

Within the restrictions described in the 3.2 section, SVM proved to be a good solution for assigned tasks.

References

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[2] John D Burger and John C Henderson. An exploration of observable features related to blogger age. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 15–20, 2006.

[3] Na Cheng, Rajarathnam Chandramouli, and KP Subbalakshmi. Author gender identification from text. *Digital Investigation*, 8(1):78–88, 2011.

[4] Katja Filippova. User demographics and language in an implicit social network. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1478–1488. Association for Computational Linguistics, 2012.

[5] Dong Nguyen, Noah A Smith, and Carolyn P Rosé. Author age prediction from text using linear regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 115–123. Association for Computational Linguistics, 2011.

[6] Claudia Peersman, Walter Daelemans, and Leona Van Vaerenbergh. Predicting age and gender in online social networks. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, pages 37–44. ACM, 2011.

[7] James W Pennebaker and Laura A King. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296, 1999.

[8] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM, 2010.

[9] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.