

Are n-gram Categories Helpful in Text Classification?

Jakub Kruczek, Paulina Kruczek and Marcin Kuta^[0000–0002–5496–6287]

Department of Computer Science,
Faculty of Computer Science, Electronics and Telecommunications,
AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Krakow, Poland,
mkuta@agh.edu.pl

Abstract. Character n-grams are widely used in text categorization problems and are the single most successful type of feature in authorship attribution. Their primary advantage is language independence, as they can be applied to a new language with no additional effort. Typed character n-grams reflect information about their content and context. According to previous research, typed character n-grams improve the accuracy of authorship attribution. This paper examines their effectiveness in three domains: authorship attribution, author profiling and sentiment analysis. The problem of a very high number of features is tackled with distributed Apache Spark processing.

Keywords: Character n-grams · Typed n-grams · Authorship attribution · Author profiling · Sentiment analysis

1 Introduction

Character n-grams are handcrafted features which widely serve as discriminative features in text categorization [2], authorship attribution [3] authorship verification [5], plagiarism detection [9, 19], spam filtering [6], native language identification of text author [8], discriminating language variety [11], and many other applications.

They also help in generating good word embeddings for unknown words, thus improving classification performance in tasks based on informal texts, where a large percentage of unknown words occurs, e.g., in sentiment analysis [1, 21]. Finally, character n-grams gave notion to character n-gram graphs [4], which found applications in topic categorization of news, blog, and twitter data, but also in automatic evaluation of document summaries.

The primary advantage of character n-grams is language independence [12], i.e., the effort of porting a feature extractor and a classifier from one language to another is negligible.

Character n-grams are recognized for their surprising degree of effectiveness in authorship attribution, outperforming content words on blog data and nearly

reaching their effectiveness on email and classic literature corpora [7]. Character n-grams have also proven to be the single most effective type of feature in authorship attribution [7]. Moreover, introduction of typed character n-grams, categories and supercategories of character n-grams have contributed to improvements in authorship attribution, compared to traditional n-grams [16].

The aim of the paper is to extend research [16] and answer the question of whether typed n-grams may be effective features in author profiling and sentiment analysis as they are in authorship attribution.

Classification on the basis of character n-grams, either typed or untyped, typically introduces a very high number of features. The solution to this problem is their distributed processing, e.g., experiments in author profiling with a large number of word n-grams as features were performed in the framework of MapReduce [10]. In [18] documents from the English language Wikipedia corpus were classified according to their topic with the newer Apache Spark framework. While the authors claim their experiments to be the first implementation of a text categorization system on Apache Spark in Python using the NLTK framework, our experiments are performed with Spark on six corpora, including approximately 150 times larger PAN-AP-13 corpus [13] with up to 8464237 features and The Blog Authorship Corpus with up to 11334188 features.

By comparison, the largest work on author profiling [17] considered larger amount of data involving 15.4 million messages and 700 million instances of words, phrases, etc.

Thus, we also examine whether the distribution of preprocessing and profile classification into smaller subtasks executed on many cores and nodes is an efficient scheme in a scenario with a high number of features, larger corpora and with the application of Spark.

2 Typed n-grams

We briefly recall the notion of typed character n-grams (in short, typed n-grams) [16]. The category and supercategory of an n-gram depends on its content and position within a word or sentence. We can distinguish between *affix*, *word* and *punct* supercategories, reflecting morpho-syntax, document topic, and author's style, respectively. Within each supercategory, we can further distinguish fine-grained categories. Within the *affix* supercategory, *prefix* and *suffix* categories denote n-grams as being the proper prefixes and proper suffixes of words, while the *space-prefix* and *space-suffix* categories denote n-grams beginning and ending with a space, respectively. Categories in the *word* supercategory (*whole-word*, *mid-word*, *multi-word*) are assigned to n-grams covering an entire word, the non-affix part of a word, or spanning multiple words, respectively. The specific category of the *punct* supercategory (*beg-punct*, *mid-punct*, *end-punct*) is assigned to n-grams containing one or more punctuation characters. Examples of some of typed n-grams arising from sentence *The actors wanted to see if the pact seemed like an old-fashioned one.* are shown in Table 1 – their detailed description can be found in [16].

Table 1. Examples of typed character n -grams of different categories for $n = 3$. Character n -grams are in red. Remaining characters (in black) denote their context. Character \square denotes space. Based on examples from [16]

Supercategory	Category	Examples
affix	prefix	actors wanted seemed
	suffix	pact actors wanted like
	space-prefix	\square actors \square wanted \square see \square like
	space-suffix	actors \square pact \square wanted \square see \square
word	whole-word	one see the
	mid-word	actors actors old-fashioned
	multi-word	actors \square wanted see \square if
punct	beg-punct	old-fashioned
	mid-punct	old-fashioned
	end-punct	old-fashioned

3 Datasets

In our experiments with n -grams we examined three problems on six datasets: authorship attribution (CCAT_50), author profiling (PAN-AP-13, Blog author gender classification data set, The Blog Authorship Corpus) and sentiment analysis (Sentiment scale dataset v1.0, Stanford Sentiment Treebank). Table 2 briefly characterizes evaluated datasets.

Table 2. Comparison of evaluated datasets

Dataset	#texts	#authors	#classes	Balanced
PAN-AP-13 (English)	500 965	283 240	6	no ¹
PAN-AP-13 (Spanish)	151 008	90 860	6	no ¹
CCAT_50	5000	50	50	yes
Blog author gender classification data set	3227	2946	2	yes
The Blog Authorship Corpus	681 288	19 320	6	no
Sentiment scale dataset v1.0	5006	4	3 and 4	no ²
Stanford Sentiment Treebank	215 154	-	5	no

¹ corpus is balanced by sex but imbalanced by age group

² Gaussian-like distribution

Figure 1 shows the proportions of categories of typed n -grams in the English part of PAN-AP-13 corpus. We can observe that together, n -grams with *multi-word* and *mid-punct* categories constitute more than half of all typed n -grams in PAN-AP-13. Figure 2 presents the number of different n -grams depending on the n -gram length. By comparison, the number of n -gram tokens in the trai-

ning, validation and test sets was approximately 1 030 960 000, 58 760 000 and 77 190 000, respectively.

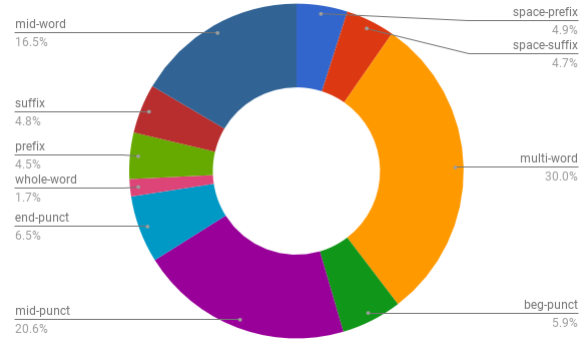


Fig. 1. Proportions of n-gram categories in the English part of PAN-AP-13 corpus

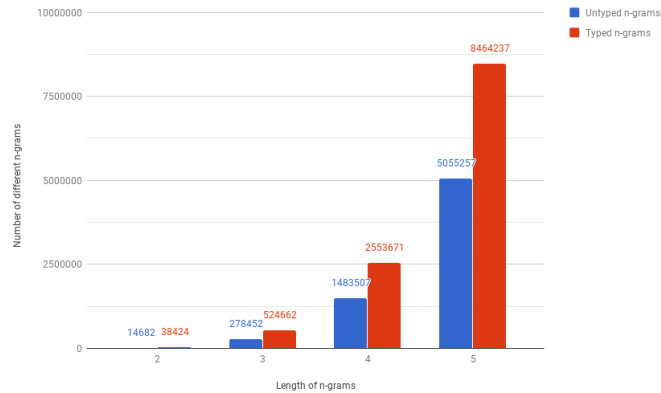


Fig. 2. Number of different character n-grams in the English part of PAN-AP-13 corpus

4 Experiments and Results

In the experiments with PAN-AP-13, corpus preprocessing involved rejecting only a few texts due to unrecognized encoding, and removing html tags and superfluous white spaces. Unknown tokens in the validation or test set were omitted.

CCAT_50 preprocessing followed the procedure from [16] and consisted of removal of citations and authors' signatures at the end of articles. Typed n-grams occurring at least five times in the dataset were taken into account as features.

Preprocessing of remaining datasets consisted of removing spurious white characters and URL addresses.

For PAN-AP-13 we adopted the predefined split into training, validation and test sets. Two classifiers were compared: multinomial Naïve Bayes (with and without feature normalization) and linear SVM based on OWLQN solver, both from Apache Spark library.

Remaining datasets were evaluated with nested cross-validation with $k = 5$ [14]. We compared three classifiers: decision trees, Naïve Bayes (multinomial and complement versions) and linear SVM, all from scikit-learn library.

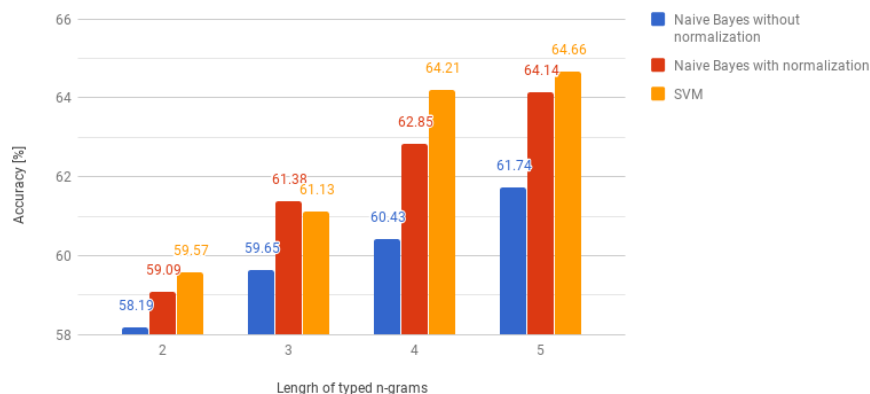


Fig. 3. Accuracy of age interval recognition depending on length of typed n-grams and obtained on the PAN-AP-13 validation set

Table 3 presents accuracy of author profile predictions for age, sex and joint profile, evaluated on the PAN-AP-13 validation set. Parameter C denotes the regularization weight in the SVM cost function, k denotes the maximal number of iterations of the SVM solver and α is the smoothing parameter in the Naïve Bayes classification. Naïve Bayes was used with n-gram normalization.

Table 4 shows corresponding accuracies of author profiling obtained on the PAN-AP-13 test set. The obtained results outperform all solutions within the PAN-AP'13 task, which often used sophisticated features of various kinds. It is interesting to compare our outcomes with the results obtained in [10]. On the same corpus, their Naïve Bayes classifier with word n-gram features achieved a profiling accuracy of 42.57%, while using conventional character n-grams as features gave only 31.20% accuracy.

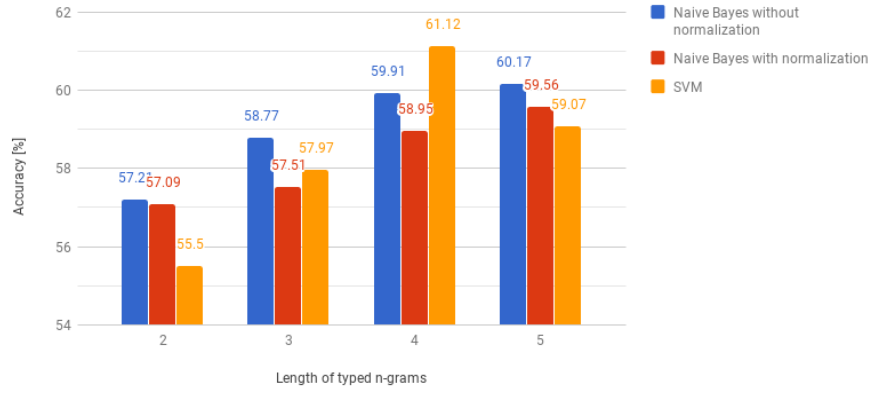


Fig. 4. Accuracy of sex recognition depending on length of typed n-grams and obtained on the PAN-AP-13 validation set

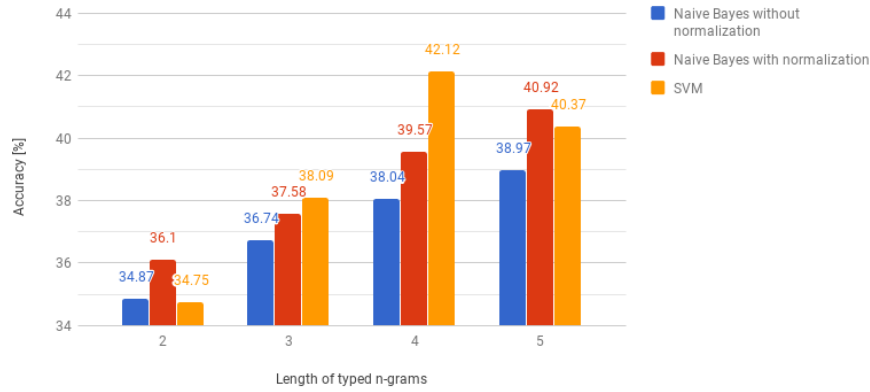


Fig. 5. Accuracy of joint profile recognition depending on length of typed n-grams and obtained on the PAN-AP-13 validation set

Table 3. Prediction accuracy of sex and age of author on the PAN-AP-13 validation set, [%]

Classifier	N-gram length	Parameters	Age	Sex	Joint profile
SVM	4-grams	$C: 500, k: 5$	64.21	61.12	42.12
SVM	4-grams	$C: 1000, k: 1$	64.44	60.68	41.59
SVM	4-grams	$C: 500, k: 1$	65.11	58.08	41.24
Naïve Bayes	5-grams	$\alpha: 1.0$	64.14	59.56	40.92
Random	–	–	33.33	50.00	16.67

Table 4. Accuracy of best models on the PAN-AP-13 test set, [%]

Classifier	N-gram length	Parameters	Age	Sex	Joint profile
SVM	4-grams	$C: 500, k: 5$	64.03	60.32	40.76
SVM	4-grams	$C: 1000, k: 1$	65.32	59.97	41.02
SVM	4-grams	$C: 500, k: 1$	65.67	57.41	40.26
SVM	4-grams	$C: 0.1, k: 5$	62.60	59.69	39.63
Naïve Bayes	5-grams	$\alpha = 1.0$	64.78	59.07	40.35

Figures 3, 4 and 5 present the accuracy of age, sex and joint recognition using typed n-grams as features, as a function of the length of used n-grams. Typed n-gram features of all categories were included in classification.

Usually, n-grams with $n = 3$ are considered in literature [16]. Our studies show that it is beneficial to consider longer n-grams with $n = 4$ or even $n = 5$. Using vargrams (e.g., 2-grams and 3-grams as one feature, not shown in figures) is not beneficial as they gave averaged results over n-grams with fixed n .

If time is not an issue, the choice of SVM over Naïve Bayes is preferred – this stays consistent with [20], advising SVM for classification of longer texts and Naïve Bayes for shorter texts.

The impact of feature normalization on Naïve Bayes is not clear; thus, no recommendation can be formulated. While it improves accuracy of age and joint profile classification, its effect on sex classification is negative. For feature scaling with SVM, standardization is always preferred over normalization [15], and it is the way in which SVM implementation from MLLib works.

Impact of n-gram categories Results in this subsection are reported for multinomial Naïve Bayes with feature normalization and size 5 n-grams. Naïve Bayes was chosen due to its better time performance over SVM. The first experiment in this part examined the impact of n-gram categories on profiling accuracy. Figures 6 and 7 shows accuracies for each of 10 categories. Additionally, classification results are shown for n-grams with no distinguished categories (*no categories*, i.e. traditional, untyped n-grams) and for features, where n-grams of *all categories* are taken into account. We observe, that compared to untyped n-grams, using whole context (all categories) increases accuracy, but the increase is tiny - 40.92% for typed n-grams vs 40.43% for untyped n-grams. Typed n-grams of any single category are worse profile predictors than untyped n-grams.

The next experiment, shown in Fig. 8, looked into the discriminative power of supercategories. Profiling accuracies obtained for *all supercategories* and *all categories* features are similar. The experiment confirms findings for categories: compared to using a single supercategory, accuracy gain achieved with *all supercategories* is tiny.

Because no single n-gram category outperformed untyped n-grams and n-grams of all categories achieved the highest accuracy, in the third experiment we considered custom categories (Fig. 9). The first custom category bundled

the four most discriminative categories and the second custom category bundled the nine most discriminative categories (i.e., all 10 categories but *whole-word*). Bundling more categories successively increases accuracy.

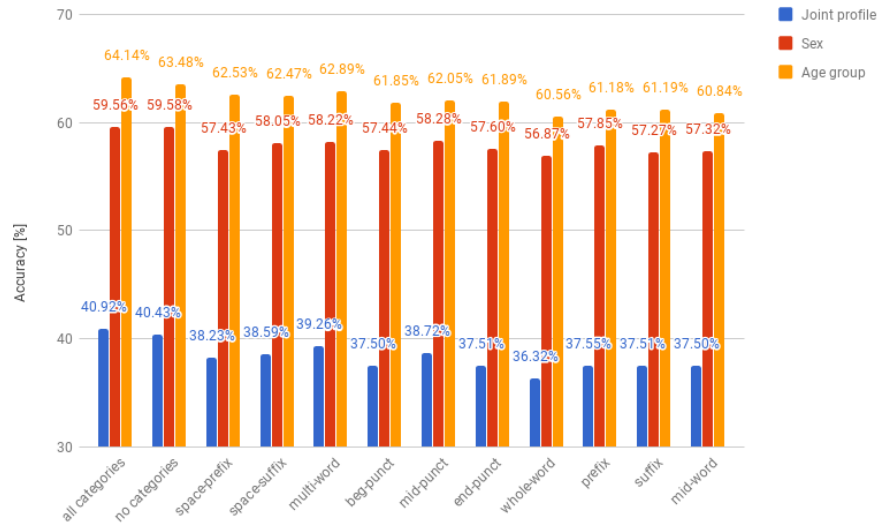


Fig. 6. Impact of n-gram categories on profiling accuracy obtained on the PAN-AP-13 validation set (English)

Impact of hyperparameters Figure 10 presents the impact of SVM hyperparameters on author profiling accuracy. Forty-five evaluations of the SVM classifier for different settings of C and k were performed. We observe that the choice of hyperparameters may impact profiling accuracy dramatically and accuracy varies from 42.12% for $(C = 5, k = 5)$ to 21.07% for $(C = 15, k = 1000)$. Choosing a good set of hyperparameters is much more important than the choice between typed and untyped n-grams in the case of the SVM classifier.

4.1 Further experiments

We performed further experiments on five datasets from Table 2. First, we performed authorship attribution experiments on CCAT_50 following setup defined in [16] (Table 5).

Table 6 presents classification accuracy on five datasets performed with untyped n-grams and all-categories typed n-grams for $n = 4$ and $n = 5$.

Throughout all datasets, in most cases typed character n-grams improve classification accuracy in comparison to untyped character n-grams. The accuracy gain is however tiny – from 0.75% to 1.48%.

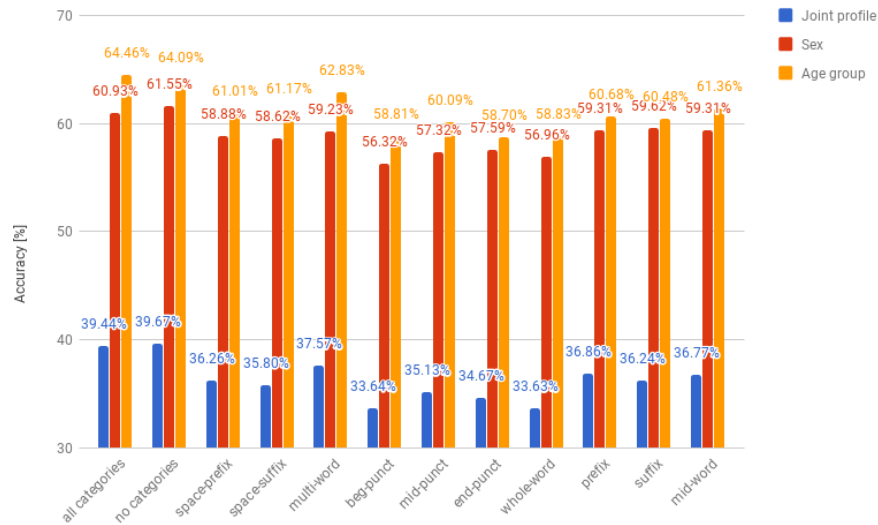


Fig. 7. Impact of n-gram categories on profiling accuracy obtained on the PAN-AP-13 validation set (Spanish)

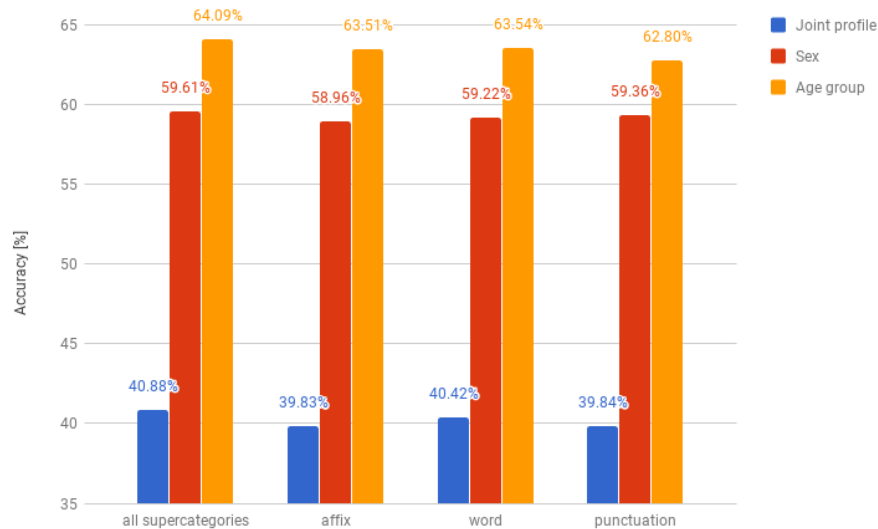


Fig. 8. Impact of n-gram supercategories on profiling accuracy obtained on the PAN-AP-13 validation set

The choice of the classifier is significant for classification with character n-grams. For all examined problems and datasets, SVM achieved higher accuracy than Naïve Bayes, with accuracy gap up to 18%.

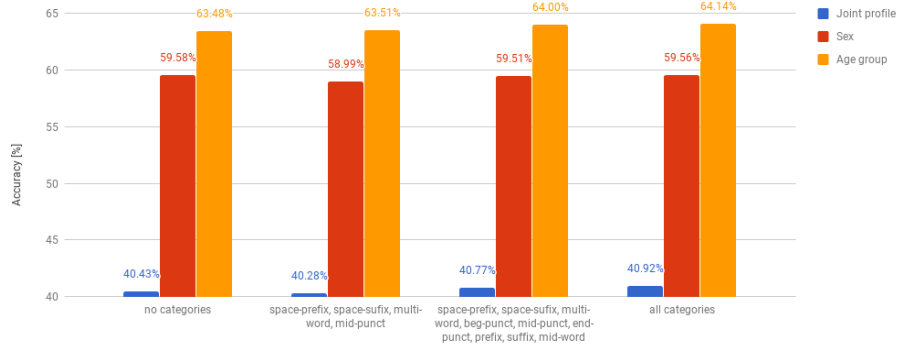


Fig. 9. Impact of custom categories of n-grams on profiling accuracy obtained on the PAN-AP-13 validation set

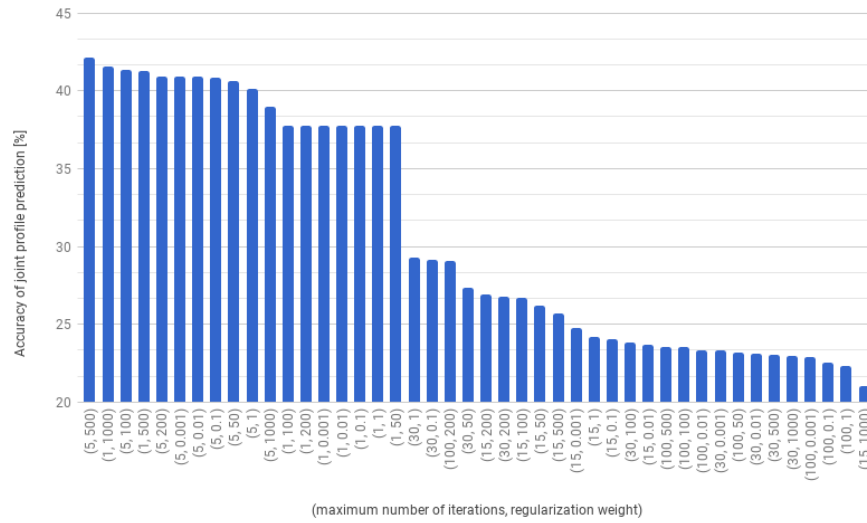


Fig. 10. Impact of SVM hyperparameters on author profiling accuracy for PAN-AP-13

We examined single-category and multiple-category n-grams. Single-category typed character n-grams differ in their predictive power w.r.t. category. Statistical tests on the Blog author gender dataset revealed that differences in accuracy are statistically significant for some pairs of categories but deeper research is needed in this area to confirm them and detect potential patterns.

Bundling more categories into typed n-grams usually results in increased accuracy. The exception was the Blog author gender classification data set, with the best results for affix+punct supercategory. Our experiments showed that

Table 5. Accuracy of authorship attribution on the CCAT_50 set, depending on used 3-gram features, [%], *acc* denotes accuracy, *N* is the number of features.

Classifier	untyped		typed		affix+punct	
	<i>acc</i>	<i>N</i>	<i>acc</i>	<i>N</i>	<i>acc</i>	<i>N</i>
SVM (Weka) [16]	69.20	14 461	69.10	17 062	69.30	9966
no tf-idf weighting						
SVM (libsvm)	84.30		84.72		82.98	
MultinomialNB	79.46	14 689	80.06	17 294	79.08	10 084
ComplementNB	71.72		70.88		70.34	
with tf-idf weighting						
SVM	84.74		85.30		85.04	
MultinomialNB	78.26	14 689	79.32	17 294	77.84	10 084
ComplementNB	73.44		73.92		72.68	

Table 6. Accuracy of untyped n-grams and all-categories typed n-grams on five datasets

Classifier	4-grams	typed 4-grams	5-grams	typed 5-grams
Blog author gender classification dataset				
SVM	71.51	71.23	70.06	70.80
MultinomialNB	67.05	67.64	68.60	68.79
ComplementNB	67.70	69.10	69.53	70.15
Blog Authorship Corpus				
SVM	62.50	62.98	63.54	64.29
MultinomialNB	46.59	46.58	47.47	47.02
ComplementNB	43.62	43.91	45.48	45.24
Sentiment scale dataset (3 classes)				
SVM	67.00	67.84	67.00	68.48
MultinomialNB	49.26	50.06	50.68	50.94
ComplementNB	51.56	50.56	49.46	49.24
Sentiment scale dataset (4 classes)				
SVM	58.09	58.37	59.19	60.79
MultinomialNB	44.35	43.61	43.53	44.51
ComplementNB	40.51	41.87	42.87	43.53
Stanford Sentiment Treebank				
SVM	59.43	60.39	60.70	61.35
MultinomialNB	60.19	60.10	60.72	60.00
ComplementNB	53.18	53.93	54.40	55.54

information about document target label is distributed among character n-grams and their categories.

Length of n-grams affects classification results and depends on the dataset and used classifier. The highest accuracy for CCAT_50 used in authorship attribution was achieved with typed 4-grams. For all remaining datasets, the best accuracy was achieved with typed 5-grams. In particular, for the Blog author gender classification dataset the highest accuracy, 71.60%, was for typed 5-grams of affix+punct supercategory (not shown in Table 6). These findings are in line with results obtained for the PAN-AP-13 corpus. Our findings clearly contradict those of [16], where authors state: *We chosen $n = 3$ since our preliminary experiments found character 3-grams to be more effective than other higher level character n-grams.*

When considering typed n-grams, the highest accuracy was achieved when bundling all categories, i.e., for all-categories typed n-grams. The only exception was the Blog author gender classification data set, where affix+punct typed n-grams achieved the highest accuracy. For all datasets, using typed character n-grams of single category results in accuracy drop in comparison to untyped character n-grams. Except for the Blog author gender classification data set, using single-supercategory n-grams resulted in lower accuracy. The best results were achieved for categories space-prefix, space-suffix, prefix, and for supercategories affix and affix+punct.

Our experiments on the Blog author gender classification dataset show that character n-grams (whether typed or untyped) give higher accuracy than word n-grams by 1%–1.15%. The downside is a larger number of arising character n-gram features than word n-gram features.

Tf-idf weighting raises classification accuracy with n-grams from 2% to 4%. The exception is authorship attribution on CCAT_50 dataset, where accuracy increased for n-grams with $n = 2$ and $n = 3$ while there was an accuracy drop for $n = 4$ and $n = 5$.

There is no clear pattern for impact of feature normalization on accuracy. The best results were obtained with normalization according to L_2 norm³. With remaining methods - StandardScaler and MaxAbsScaler we observed suboptimal accuracy or even accuracy worse than with no normalization.

Finally, we performed qualitative analysis and looked for the most important n-grams by inspecting weights of SVM classifier. First, we analysed author profiling on the Blog author gender classification data set. For men, identified n-grams referred to wife, other men (guys) and games. The most important n-grams used by women are related to family (love, husband, mum). Found best n-grams do not suggest that text style (e.g. punctuation) is important for a classifier. Next, we analysed authorship attribution on one particular author chosen from CCAT_50: identified n-grams were name fragments of cites, states or companies.

³ Normalizer from scikit-learn library

5 Conclusions

The paper has shown in three domains: authorship attribution, author profiling and sentiment analysis that the choice of typed n-grams results in only a tiny increase of classification accuracy over traditional n-grams. Information about the author profile is distributed throughout all n-gram categories. No single category can be advised for classification. It is worth putting much more effort into effective hyperparameter optimization and model selection than to switching from n-grams to typed n-grams or particular category of typed n-grams.

Apache Spark allows for efficient classification with a very high number of features on large text corpora. The memory footprint is the most prohibitive aspect of such classification, which precludes experiments with n-grams longer than 5.

Acknowledgements. The research presented in this paper was supported by the funds assigned to AGH University of Science and Technology by the Polish Ministry of Science and Higher Education. This research was supported in part by PL-Grid Infrastructure.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**, 135–146 (2017)
2. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval, SDAIR-94*. pp. 161–175 (1994)
3. Escalante, H.J., Solorio, T., Montes-y-Gómez, M.: Local histograms of character n-grams for authorship attribution. In: Lin, D., Matsumoto, Y., Mihalcea, R. (eds.) *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. pp. 288–298 (2011)
4. Giannakopoulos, G., Karkaletsis, V.: N-gram graphs: Representing documents and document sets in summary system evaluation. In: *Proceedings of the Second Text Analysis Conference, TAC 2009*. NIST (2009)
5. Jankowska, M., Milios, E.E., Keselj, V.: Author verification using common n-gram profiles of text documents. In: Hajic, J., Tsujii, J. (eds.) *25th International Conference on Computational Linguistics, COLING 2014*. pp. 387–397 (2014)
6. Kanaris, I., Kanaris, K., Houvardas, I., Stamatatos, E.: Words versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools* **16**(6), 1047–1067 (2007)
7. Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology* **60**(1), 9–26 (2009)
8. Koppel, M., Schler, J., Zigdon, K.: Automatically determining an anonymous author’s native language. In: Kantor, P.B., Muresan, G., Roberts, F.S., Zeng, D.D., Wang, F., Chen, H., Merkle, R.C. (eds.) *Intelligence and Security Informatics, IEEE International Conference on Intelligence and Security Informatics, ISI 2005*. pp. 209–217 (2005)

9. Kuta, M., Kitowski, J.: Optimisation of Character n-gram Profiles Method for Intrinsic Plagiarism Detection. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) 13th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2014. Lecture Notes in Artificial Intelligence, vol. 8468, pp. 500–511 (2014)
10. Maharjan, S., Shrestha, P., Solorio, T., Hasan, R.: A Straightforward Author Profiling Approach in MapReduce. In: Bazzan, A.L.C., Pichara, K. (eds.) Advances in Artificial Intelligence - IBERAMIA 2014 - 14th Ibero-American Conference on AI. pp. 95–107 (2014)
11. Malmasi, S., Dras, M.: Language identification using classifier ensembles. In: Nakov, P., Zampieri, M., Osenova, P., Tan, L., Vertan, C., Ljubešić, N., Tiedemann, J. (eds.) Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects. pp. 35–43. Association for Computational Linguistics (2015)
12. Peng, F., Schuurmans, D., Keselj, V., Wang, S.: Language independent authorship attribution with character level n-grams. In: 10th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2003. pp. 267–274 (2003)
13. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the Author Profiling Task at PAN 2013. In: Forner, P., Navigli, R., Tufis, D., Ferro, N. (eds.) Working Notes for CLEF 2013 Conference. vol. 1179 (2013)
14. Raschka, S.: Model evaluation, model selection, and algorithm selection in machine learning. CoRR **abs/1811.12808** (2018)
15. Raschka, S., Mirjalili, V.: Python Machine Learning. Packt Publishing, 2 edn. (2017)
16. Sapkota, U., Bethard, S., Montes-y-Gómez, M., Solorio, T.: Not all character n-grams are created equal: A study in authorship attribution. In: Mihalcea, R., Chai, J.Y., Sarkar, A. (eds.) NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 93–102 (2015)
17. Schwartz, H.A., Eichstaedt, J.C., Kern, M.L., Dziurzynski, L., Ramones, S.M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M.E.P., Ungar, L.H.: Personality, gender, and age in the language of social media: The open-vocabulary approach. PLoS ONE **8**(9), 1–16 (2013). <https://doi.org/10.1371/journal.pone.0073791>
18. Semberecki, P., Maciejewski, H.: Distributed classification of text documents on Apache Spark platform. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) 15th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2016. pp. 621–630 (2016)
19. Stamatatos, E.: Intrinsic plagiarism detection using character n-gram profiles. In: Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.) SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009). pp. 38–46 (2009)
20. Wang, S.I., Manning, C.D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: Li, H., Lin, C.Y., Osborne, M., Lee, G.G., Park, J.C. (eds.) 50th Annual Meeting of the Association for Computational Linguistics. pp. 90–94 (2012)
21. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Charagram: Embedding words and sentences via character n-grams. In: Su, J., Carreras, X., Duh, K. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016. pp. 1504–1515 (2016)