Team aida at PAN: Ensembling Normalized Log Probabilities

Notebook for the PAN Lab at CLEF 2024

Pablo Miralles[1], Alejandro Martín[1] and David Camacho[1]

[1] Technical University of Madrid, Alan Turing st. 28031, Madrid, Spain

Abstract

The advent of LLMs has revolutionized the way we produce text, greatly increasing our productivity. However, dual uses such as misinformation generation or human impersonation have also become a concern, making it desirable to be able to automatically distinguish between human and AI-generated text. In this paper, we present an ensemble method to detect AI-generated based on how suprising the text is for a group of open LLMs.

Keywords

AI-generated text detection, LLMs, Log probabilities, Ensemble,

1. Introduction

The rise of Large Language Models (LLMs) is a new technological revolution, offering the ability to generate human-like text at an unprecedented scale. While this promises great increases in productivity, it also raises concerns about the potential for misuse. In particular, LLMs can be used to generate fake news, impersonate people, or spread misinformation. As a result, there is a growing need for tools that can detect AI-generated text. In this paper, we propose an approach to detect AI-generated text based on the distribution of token probabilities of multiple open LLMs. This approach is submitted to the Voight-Kampff Generative AI Authorship Verification 2024 task [1] at PAN [2].

2. Background

It is well-known that LLMs generate text one token at a time, predicting the next token based on the previous ones. To do this, the model outputs a distribution of probabilities over the vocabulary and the next token is sampled from this distribution using a fixed strategy.

Applying an LLM to a full text results in a sequence of probability distributions, one for each token. If the text is generated by that LLM, we expect the following token to have a high probability in the conditional distribution generated by all the previous ones. On the other hand, human text is expected to contain more statistical outliers, and be more "surprising" to the model on average. This idea has been used in the past, in works such as Ippolito et al. [3] and Gehrmann et al. [4]. They used the raw probabilities of the LLMs as well as the rank of the target token in the probability distribution.

This method presents a series of problems.

1. First, we do not know or do not have access to the exact model that generated the text. Although we expect most models to be pretrained on a similar large corpus of text, the choice of vocabulary or fine-tuning data and procedure might affect the probabilities.
2. Second, we do not have access to the prompt that was used, which strongly conditions the generation of the text. Without the prompt, the perplexity of the text will naturally be higher, as the model does not see the context that conditioned it. This is especially the case if the prompt

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

pablo.miralles@upm.es (P. Miralles); alejandro.martin@upm.es (A. Martín); david.camacho@upm.es (D. Camacho)

0000-0002-1230-9035 (P. Miralles); 0000-0002-0800-7632 (A. Martín); 0000-0002-5051-3475 (D. Camacho)

forces the model to generate text about an unusual topic or in an unusual style. This problem was already pointed out by Hans et al. [5].

3. Third, some parts of the text allow for more randomness than others. For example, the beginning of a text or a sentence allows for more variability than finishing a word that was started in the previous token. As another example, the sentence “My favorite color is” allows any color to be the next token, and the model will have to distribute the probability across several tokens.

Using the rank of the target token in the probability distribution already hinted at the third problem, although very unconditioned tokens might still yield high ranks.

Hans et al. [5] try to solve this problem by normalizing the log-probabilities. First, they used an LLM M_1 to calculate the mean log-probabilities across all tokens of the text. Then, they used a second LLM M_2 to calculate the mean cross-entropy of the distributions given by M_1 and M_2 for each token of the text. Finally, they normalize the mean log-probabilities by the mean cross-entropy.

If we consider using the same LLM $M_1 = M_2$, then the entropy values measure how much randomness there is in the distribution that generated the token. Normalizing by this value reduces the importance of having a low probability in a high entropy distribution, helping with the second and third problem. In practice, the authors used very similar LLMs, such as the foundation and instruct versions of the same model—perhaps because the tokenizer needed to be identical—. We expect distributions to be close and this interpretation to be valid.

Another similar idea is that of Bao et al. [6]. They compare the mean log-probabilities with the mean log-probabilities of a large number of resampled texts. New texts are generated by sampling each token from the distributions of the original text, independently. This can be seen as a Monte Carlo approximation of the Shannon entropy of the distribution.

3. System Overview

3.1. Conceptual approach

Our method consists of three main steps. First, we compute a metric derived from the log-probabilities output by an LLM, trying to solve the second and third problems discussed in the background. For each LLM, we get a distribution of this metric across the full sequence. Second, we extract a set of features from these distributions, as well as some classic NLP features from the text. Finally, we use these features to train a classifier that predicts whether the text was generated by an LLM or not.

3.1.1. Normalized log-probability

We use a metric that takes into account both the probability of the next token and how randomly the probability is distributed. Consider a sequence of tokens $t_1, \ldots, t_M$. After passing through an LLM, we get a matrix of logits $L \in \mathbb{R}^{M \times V}$, where V is the vocabulary size. Then, for all m from 1 to $M - 1$, we can compute a normalized log-probability

$$\mathrm{NLP}_m = \left(\max_{v \in \{1, \ldots, V\}} L_{m,v} \right) - L_{m,t_{m+1}}.$$

This metric compares the likelihood of the token t_{m+1} with the likelihood of the most likely token. If the token is the most likely one, then the value will be zero. If the token is very unlikely, then it will only be penalized if there are tokens that are much more likely.

3.1.2. Extracting features

The previous procedure yields, for a given text and LLM, a sequence of normalized log-probability values $\mathrm{NLP}_1, \ldots, \mathrm{NLP}_{M-1}$. We propose to model it as a set of observations from a distribution. We can then extract features such as the mean, the variance, or the quantiles of this distribution. We also extract some features from seeing it as an ordered sequence. After getting these features, we concatenate them for several LLMs. Further, we add some classic NLP features such as the ratio of stop words, the ratio of punctuation, the ratio of each syntactic category...

3.1.3. Modeling

Finally, we use an XGBoost classifier to predict whether the text was generated by an LLM or not based on the features.

3.2. Submission hyperparameters and features

We use the following models, given their relevance in the state-of-the-art literature and our computational resources:

- `google/gemma-2b-it` [7]
- `meta-llama/Llama-2-7b-chat-hf` [8]
- `mistralai/Mistral-7B-Instruct-v0.2` [9]
- `databricks/dolly-v2-3b` [10]
- `stabilityai/stablelm-zephyr-3b` [11]
- `microsoft/phi-1_5` [12]

For each model and its respective values $\mathrm{NLP}_1, \ldots, \mathrm{NLP}_{M-1}$, we extract the following features: the mean, the standard deviation, the minimum value, the ratio of values that are close to zero, and the ratio of values where the sequence is increasing.

We also extract the following classic NLP features:

- Ratio of verbs, nouns, adjectives, adverbs, personal pronouns, prepositions or subordinating conjunctions, coordinating conjunctions, determiners, interjections, particles, and cardinal numbers.
- Ratio of words of other types.
- Number of words.
- Mean sentence length.

We tune the XGBoost hyperparameters using cross-validation, and train using only the given dataset. As the dataset is unbalanced, we use oversampling for training and balanced accuracy as validation metric.

3.3. Task predictions

Finally, we train for raw classification, but TIRA [13] submissions require distinguishing, for a given pair of texts, which text is AI-generated. Thus, we need to transform our predictions. If we get probabilities of being human p_1 and p_2 from the first and second texts, respectively, we output $\frac{p_2 - p_1 + 1}{2}$.

4. Results

We present the results of our model across cross-validation sets in the binary classification task of detecting if a given text is AI-generated or not. We compare our model against two baselines: Binoculars [5] and Fast-DetectGPT [6]. The results are shown in table 1. We can see that our model outperforms both baselines in terms of balanced accuracy, with a good margin.

Table 2 shows the results, initially pre-filled with the official baselines provided by the PAN organizers and summary statistics of all submissions to the task (i.e., the maximum, median, minimum, and 95-th, 75-th, and 25-th percentiles over all submissions to the task). In this case, the task is to distinguish which of two given texts is generated by AI. The results show that our model outperforms the baselines, although with a small margin. Our model appears to be outperformed by other solutions, except in the

ROC-AUC metric. Further discussion could be provided with more knowledge of the test sets and other solutions.

Table 3 shows the summarized results averaged (arithmetic mean) over 10 variants of the test dataset. Each dataset variant applies one potential technique to measure the robustness of authorship verification approaches, e.g., switching the text encoding, translating the text, switchign the domain, manual obfuscation by humans, etc.

Table 1
Balanced accuracy of our method against two baselines in the validation set. In this case, we consider a binary classification problem where the model predicts if the text is AI-generated or not. For Binoculars, we use `tiiuae/falcon-7b` as the observer model, and `tiiuae/falcon-7b-instruct` as the performer model. For Fast-DetectGPT, we use the model `tiiuae/falcon-7b`. Total accuracy is calculated through cross-validation, except for Binoculars which was pretrained.

Aproach	Balanced accuracy
Our method	0.957
Binoculars [5]	0.852
Fast-DetectGPT [6]	0.423

Table 2
Overview of the accuracy in detecting if a text is written by an human in task 4 on PAN 2024 (Voight-Kampff Generative AI Authorship Verification). We report ROC-AUC, Brier, C@1, F_1, $F_{0.5u}$ and their mean.

Approach	**ROC-AUC**	**Brier**	**C@1**	**F_1**	**$F_{0.5u}$**	**Mean**
vicious-artifact	0.995	0.954	0.976	0.976	0.975	0.975
corporate-burn	0.995	0.954	0.976	0.976	0.976	0.976
Baseline Binoculars	0.972	0.957	0.966	0.964	0.965	0.965
Baseline Fast-DetectGPT (Mistral)	0.876	0.8	0.886	0.883	0.883	0.866
Baseline PPMd	0.795	0.798	0.754	0.753	0.749	0.77
Baseline Unmasking	0.697	0.774	0.691	0.658	0.666	0.697
Baseline Fast-DetectGPT	0.668	0.776	0.695	0.69	0.691	0.704
95-th quantile	0.994	0.987	0.989	0.989	0.989	0.990
75-th quantile	0.969	0.925	0.950	0.933	0.939	0.941
Median	0.909	0.890	0.887	0.871	0.867	0.889
25-th quantile	0.701	0.768	0.683	0.657	0.670	0.689
Min	0.131	0.265	0.005	0.006	0.007	0.224

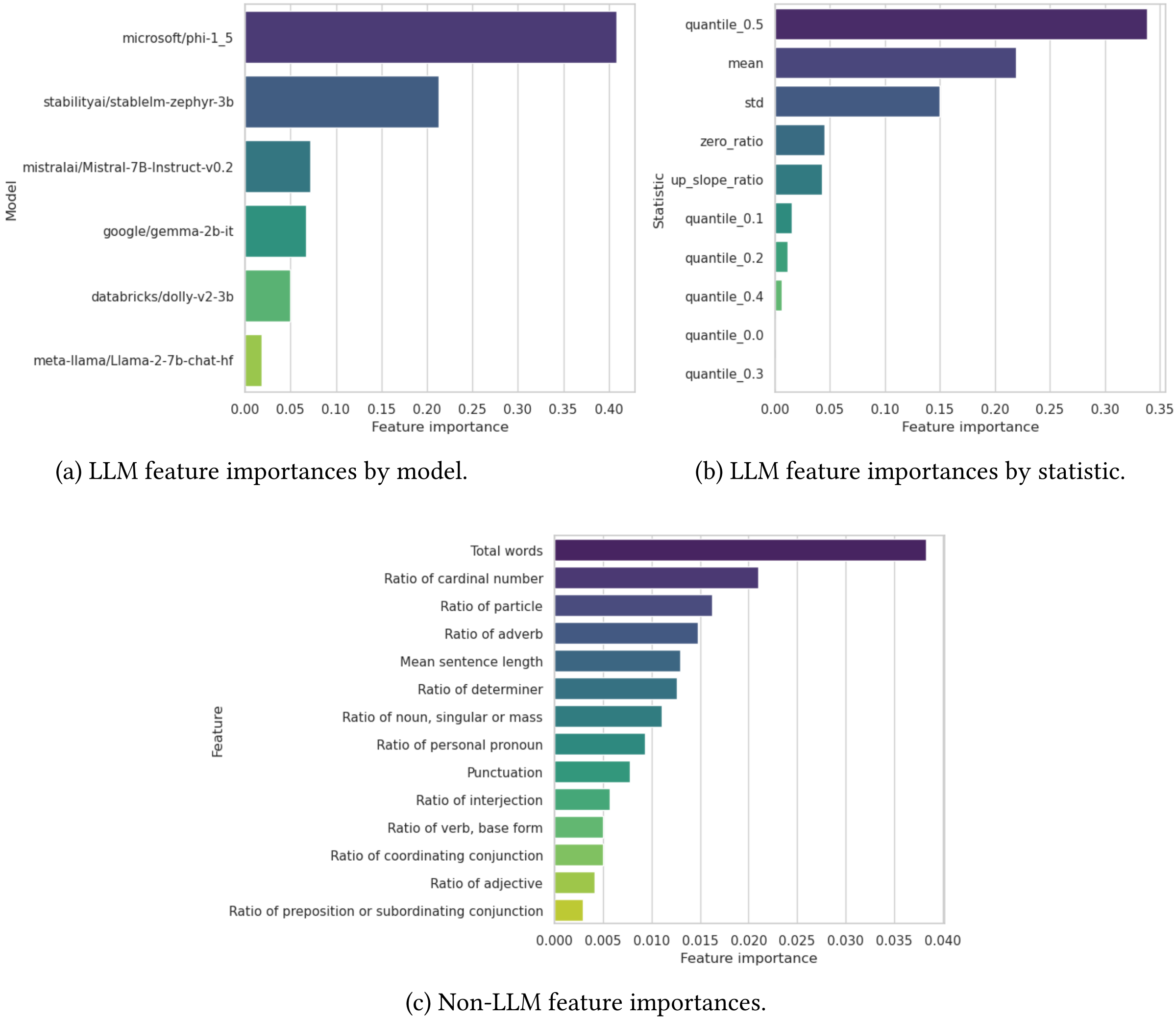
Figure 1 shows the feature importances of the model. We can see that the LLM characteristics are the most relevant, with all NLTK features having an importance lower than 0.05. The most used LLM is `phi-1_5`, although this varies greatly when rotating different LLMs. In general, any given LLM gets a good accuracy on its own. The statistics of the normalized log-probabilities that appear to be the most important are the median, the mean and the standard deviation, followed by the ratio of zero values—that is, the ratio of times where the most probable token was selected—, and the ratio of positively sloped values. This last statistic could be interesting: it measures how often the text becomes more or less predictable after one token.

5. Conclusions

We present a method to detect AI-generated text based on a small improvement to the log-probability metric. We ensemble the predictions from several LLMs, as well as classic NLP features. Our method achieves promising results, but further study on the generalization of the method is needed, as well as a proper ablation study.

Table 3
Overview of the mean accuracy over 9 variants of the test set. We report the minumum, median, the maximum, the 25-th, and the 75-th quantile, of the mean per the 9 datasets.

Approach	Minimum	25-th Quantile	Median	75-th Quantile	Max
vicious-artifact	0.696	0.837	0.931	0.975	0.994
corporate-burn	0.697	0.837	0.931	0.976	0.994
Baseline Binoculars	0.342	0.818	0.844	0.965	0.996
Baseline Fast-DetectGPT (Mistral)	0.095	0.793	0.842	0.931	0.958
Baseline PPMd	0.270	0.546	0.750	0.770	0.863
Baseline Unmasking	0.250	0.662	0.696	0.697	0.762
Baseline Fast-DetectGPT	0.159	0.579	0.704	0.719	0.982
95-th quantile	0.863	0.971	0.978	0.990	1.000
75-th quantile	0.758	0.865	0.933	0.959	0.991
Median	0.605	0.645	0.875	0.889	0.936
25-th quantile	0.353	0.496	0.658	0.675	0.711
Min	0.015	0.038	0.231	0.244	0.252



(a) LLM feature importances by model.

(b) LLM feature importances by statistic.

(c) Non-LLM feature importances.

Figure 1: Feature importances of the model.

Future work is needed to study the best way to normalize the log-probabilities to account for the problems we have discussed. Further, we should also take into account that human text is more likely to take the LLM into unchartered territory, statistically speaking. This problem and the previously discussed ones conflict, and balancing them might be needed.

Finally, one can also study the interaction between the predictions of different LLMs. We believe that having many LLMs with a diversity of training data, training procedures and vocabularies is helpful. Studying which LLMs are more likely to agree could help selecting a minimal ensemble of models, reducing the computational cost of the method.

Acknowledgments

This work has been funded by the project PCI2022-134990-2 (MARTINI) of the CHISTERA IV Cofund 2021 program, funded by MCIN/AEI/10.13039/ 501100011033 and by the "European Union NextGenerationEU/PRTR"; by the Spanish Ministry of Science and Innovation under FightDIS (PID2020-117263GB-I00); by MCIN/AEI/10.13039/501100011033/ and European Union NextGenerationEU/PRTR for XAI-Disinfodemics (PLEC2021-007681) grant. This publication has been also written in the context of Iberifier Plus project, co-funded by the European Commission under the Call DIGITAL-2023-DEPLOY-04, European Digital Media observatory (EDMO) – National and multinational hubs; reference: IBERIFIER Plus - 101158511.

References

[1] J. Bevendorff, M. Wiegmann, J. Karlgren, L. Dürlich, E. Gogoulou, A. Talman, E. Stamatatos, M. Potthast, B. Stein, Overview of the "voight-kampff" generative AI authorship verification task at PAN and ELOQUENT 2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[2] A. A. Ayele, N. Babakov, J. Bevendorff, X. B. Casals, B. Chulvi, Daryna Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, D. Moskovskiy, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, N. Rizwan, P. Rosso, F. Schneider, A. Smirnova, E. Stamatatos, E. Stakovskii, B. Stein, M. Taulé, D. Ustalov, X. Wang, M. Wiegmann, S. M. Yimam, Eva Zangerle, Overview of PAN 2024: Multi-author writing style analysis, multilingual text detoxification, oppositional thinking analysis, and generative AI authorship verification, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[3] D. Ippolito, D. Duckworth, C. Callison-Burch, D. Eck, Automatic Detection of Generated Text is Easiest when Humans are Fooled, 2020. arXiv:1911.00650.

[4] S. Gehrmann, H. Strobelt, A. M. Rush, GLTR: Statistical Detection and Visualization of Generated Text, 2019. arXiv:1906.04043.

[5] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text, 2024. arXiv:2401.12070.

[6] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang, Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature, 2024. arXiv:2310.05130.

[7] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikuła, M. Wirth,

M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, P. G. Sessa, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y.-h. Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, K. Kenealy, Gemma: Open Models Based on Gemini Research and Technology, 2024. arXiv:2403.08295.

[8] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023. arXiv:2307.09288.

[9] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7B, 2023. arXiv:2310.06825.

[10] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, R. Xin, Free dolly: Introducing the world's first truly open instruction-tuned LLM, https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm, 2023.

[11] Stabilityai/stablelm-zephyr-3b · Hugging Face, https://huggingface.co/stabilityai/stablelm-zephyr-3b, 2024.

[12] Y. Li, S. Bubeck, R. Eldan, A. Del Giorno, S. Gunasekar, Y. T. Lee, Textbooks Are All You Need II: Phi-1.5 technical report, 2023. arXiv:2309.05463.

[13] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous integration for reproducible shared tasks with tira.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.