TNA ACTIVITY REPORT

Parametrization Guidelines in StyloR for the Romanian Language

Author: Maria-Corina Dimitriu

Current position: MA student, young researcher

Affiliation: “Alexandru Ioan Cuza” University of Iași, Romania

Host institution: Trier Center for Digital Humanities, University of Trier

Mentor(s): Prof. Christof Schöch

Period of stay: 22.10.2024 – 06.11.2024

Introduction of the Project

The main objective of my project was to test the performance of the StyloR package for intuitive users when applied to the Romanian language and to see whether there are sets of parameters that give optimal results for different types of stylometric analysis. In the context of applying for a five-week fellowship, my overachieving aim was to elaborate a written guide for using this tool in Romanian and for the Romanian language, so that more students from my country would use StyloR in their research. Since I only received eleven days of fellowship, I decided to keep this idea as a long-range objective, and, during these eleven days, I focused on enhancing the effectiveness of stylometric analysis on Romanian texts, by testing various combinations of parameters and by creating some tools that would compensate for at least some of the drawbacks of not having Romanian as a language option of the package.

Stay at the Research Infrastructure

I applied for a fellowship at the Trier Center for Digital Humanities (University of Trier) taking into account the consistent and valuable activity of this research center and my mentor in the field of stylometry. My high expectations were not only met, but also exceeded, since I came to Trier as a very basic user of StyloR and I am now able to employ quite a few functions of the package, as well as to write code for the more complex operations involved in text pre-processing and multiple tests stages. Communication with my mentor, Prof. Christof Schöch, was very efficient during my entire stay here, and he found the time to help me each time I experienced problems with my codes. Beyond the advice he gave me concerning specific issues related to my project, he showed me many useful articles and websites, that I will certainly use in my further research. Moreover, as an MA student who does not study Digital Humanities at the university (because, unfortunately, there is no such MA programme at the

CLS INFRA has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101004984

university center that I come from), I really enjoyed attending some of my mentor's introductory lectures on Linked Open Data. In the end, my visit was not only about developing my project, but also about the opportunity to learn from a differently organised and better-prepared academic community when it comes to computational literary studies.

Methodology and Implementation

For my project, I worked with the POP-LITE corpus, comprising one hundred Romanian popular novels from the long 19th century, available as text files. Since five of the novels in the corpus were written by anonymous, my analysis focused especially on authorship attribution issues, so that, while trying to optimize the authorship attribution process for the Romanian language, I may also find some stylometric evidence for the identity of the authors of those texts.

In the beginning, I renamed the files according to the template AuthorName_Title. After some preliminary tests using Cluster Analysis, Principal Components Analysis and Bootstrap Consensus Tree as methods of analysis, keeping the Classic Delta distance due to its stability, and varying the number of most frequent words (from 100 to 1000, with 100 incrementation), I decided to eliminate all the novels shorter 6500 words, being left with 92 novels. Likewise, I eliminated the five anonymous novels, with the intention of reintroducing them in the corpus once I would the most effective combinations of parameters.

In parallel, since the option of automatically deleting pronouns is not available for the Romanian language but is highly recommended by some researchers in the authorship attribution tests, I started elaborating a list of all the Romanian pronouns, which I intended to use as a custom stopwords list in further analysis. When working on the list, I included all ten categories of Romanian pronouns (personal pronouns, polite personal pronouns, reflexive pronouns, demonstrative pronouns, possessive pronouns, indefinite pronouns, negative pronouns, interrogative pronouns, relative pronouns and intensive pronouns) and grouped the pronouns by category, not only to be able to cull only some specific categories of pronouns (i.e. personal pronouns) if that proves more effective at some point, but also to make it easier for anyone to notice the possible absence of a pronoun and to complete the list accordingly. I included both the long and the short forms of the pronouns and also took into account the possible misspellings arising from the presence of Romanian special characters. For instance, in the case of a pronoun such as *atâția*, I included four forms in the list: *atâția*, *atâtia*, *atația* and *atatia*. These precautions, added to the very large number of Romanian pronouns, the existence of oral and regional forms (which do appear in my corpus) and the grammatical categories that influence the form of the pronouns (person, gender, number, case) led to a list of 1147 pronouns. Although the list has been double-checked and elaborated with constant reference to Romanian dictionaries, it would be no surprise if it gets even larger after I discuss it with my Romanian research fellows.

One specific problem in the case of Romanian pronouns is that some of the forms overlap with the forms of other Romanian functional words. For instance, the feminine 3rd person singular accusative short form *o*, as in *O văd* ("I see *her*") is the same as the feminine indefinite article *o*, as in *Văd o pasăre* ("I see *a* bird"). Since the differentiation of these homonymous forms would be very complicated and since some researchers even talked about culling not only pronouns, but also other functional words which are rather tied to the structure of a certain language than to the actual option of an author, I decided to keep these forms in my list.

Another setback that I encountered was related to the implementation of the list. Since I was working with a custom and not a pre-defined stopwords list, the program did not allow me to simply implement it as a stopwords list. Therefore, I opted for an additional step of text

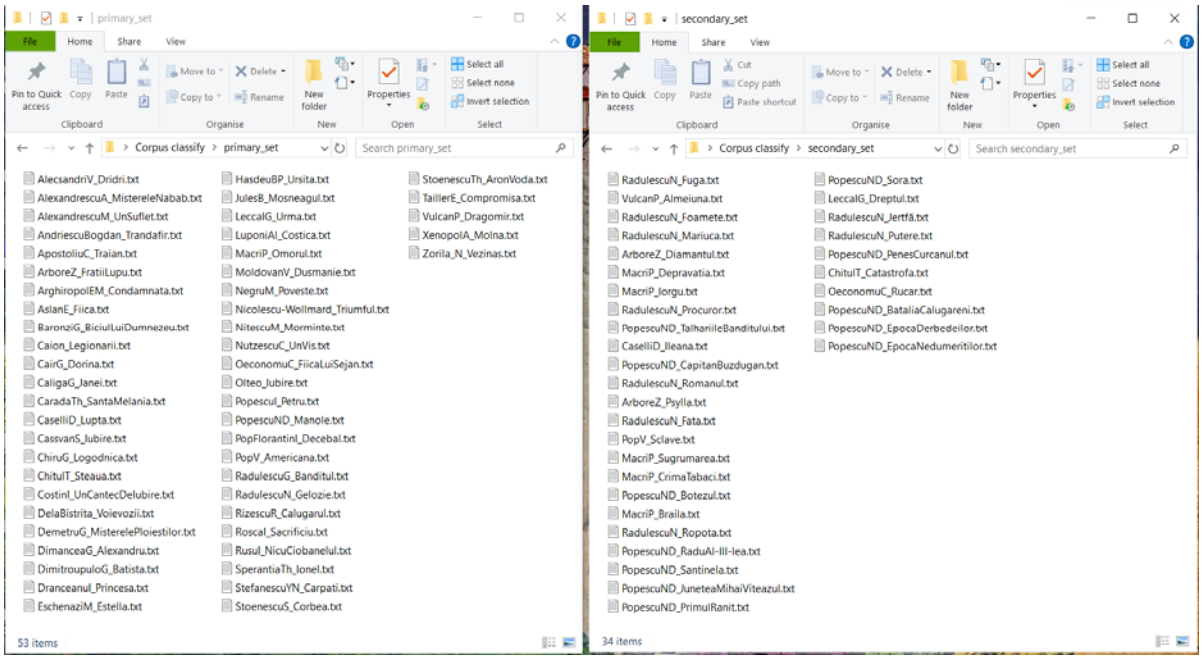
preprocessing, and I automatically created another version of the corpus, with all the pronouns deleted from the text. For this, I wrote a function that would eliminate the words in the list from a text (taking into account punctuation issues as well) and I iteratively applied it to the texts in my corpus. For each of the text in my original corpus, I created a copy with the same file name and I put all this processed copies in another directory. The lines that I used were the following:

```
stopwords_list <- readLines("Liste/stopwords_pronume.txt")
remove_stopwords <- function(text, stopwords)
{words <- unlist(strsplit(tolower(text), "\\s+"))
words <- words[!words %in% stopwords]
return(paste(words, collapse = " "))
}
files <- list.files(path = "Corpus", pattern = "\\.txt$", full.names = TRUE)
Corpus_data <- lapply(files, readLines)
processed_corpus <- sapply(Corpus_data, remove_stopwords, stopwords = stopwords_list)
dir.create("processed_corpus", showWarnings = FALSE)
for (i in seq_along(processed_primary)){
writeLines(processed_corpus[i], paste0("processed_corpus/text", i, ".txt"))
}
files <- list.files("Corpus", full.names = TRUE)
processed_files <- list.files("processed_Corpus", full.names = TRUE)
list.files(path = "Corpus", pattern = "\\.txt$", full.names = TRUE)
Corpus_filenames <- basename(files)
for (i in seq_along(processed_files)) {
file.rename(from = processed_files[i],
to = file.path("processed_corpus", Corpus_filenames[i]))
}
```

Having both the original and the processed version of the corpus proved very useful in the end. Since I had to do a lot of tests, eliminating one step (the introduction of the stopwords list) from each iteration made the code run more smoothly.

After solving the problem related to the pronoun stopwords list by creating a new folder with the processed texts, I continued by doing tests on the processed corpus. Since no combination of parameters seemed to lead to a perfect clustering of all the texts belonging to the same authors, as I had initially hoped would happen, I realised that I needed a more objective way of evaluating the effectiveness of each test and, at the advice of my mentor, I switched from the stylo() to the classify() function.

Consequently, I created two directories. In the training set (primary_set), which comprised 53 novels, I included the novels of the authors who only had one novel in the corpus, as well as a randomly chosen novel from each of the other authors. The other novels were included in the test set (secondary_set), which consisted of 34 novels. The results were not those expected and the higher accuracy rates I could obtain with this initial division were around 60%. Therefore, after reading more about the classify() function methodology, I decided to change the strategy and use representative novels of the authors in the training set. To do that, I consulted the information provided by literary history about the novels in my corpus. Lacking precise information about print runs, number of copies and distribution, my choice of representative novels was finally based on the form in which the novel was published (presuming, for instance, that a novel that first appeared in feuilleton and was also published in print afterwards was of a higher impact than a novel that remained in feuilleton), on the number of critical mentions, as well as on some explicit mentions concerning the novel's representative character, where available.



Having the novels re-arranged into the two sets based on the previously mentioned criteria, I proceeded to systematic tests. I used Delta as a classification method and I varied the type of most frequent features (words, word 2-grams, word 3-grams, character 2-grams, character 3-grams, character 4-grams, character 5-grams), the number of most frequent features (from 100 to 1000 most frequent features, with 100 incrementation) and the Delta distance (all 10 Delta distances available in the Stylo package). When working with words, word 2-grams and word 3-grams, I used both the original and the processed corpus (with the pronouns deleted). Whereas there was no test to attribute all the novels correctly and the pronoun stopwords list did not make much difference, I got some promising accuracy rates between 75% and 80%, especially when I used words and characters 5-grams as units of measure. I also noticed that the Cosine Delta distance, although not always effective, gave the most consistent results when applied to a large number of most frequent features (700 MFF or more).

	Classic Delta	Cosine Delta	Eder's Delta	Eder's simple	Entropy	Manhattan	Canberra	Euclidean	Cosine	Min-Max
100 MFW	61.8	58.8	64.7	58.8	55.9	61.8	61.8	55.9	50	55.9
200 MFW	67.6	67.6	67.6	67.6	67.6	70.6	70.6	58.8	50	70.6
300 MFW	73.5	70.6	67.6	70.6	67.6	70.6	70.6	55.9	52.9	70.6
400 MFW	70.6	67.6	73.5	73.5	67.6	64.7	76.5	52.9	47.1	73.5
500 MFW	70.6	67.6	73.5	73.5	70.6	67.6	70.6	52.9	47.1	73.5
600 MFW	73.5	70.6	73.5	73.5	73.5	67.6	70.6	47.1	41.2	73.5
700 MFW	70.6	76.5	73.5	73.5	73.5	67.6	70.6	50	50	70.6
800 MFW	73.5	79.4	73.5	73.5	73.5	70.6	67.6	50	50	70.6
900 MFW	70.6	76.5	73.5	73.5	73.5	73.5	64.7	50	50	70.6
1000 MFW	73.5	70.9	73.5	73.5	73.5	73.5	67.6	50	50	73.5

Words, no stopwords list

	Classic Delta	Cosine Delta	Eder's Delta	Eder's simple	Entropy	Manhattan	Canberra	Euclidean	Cosine	Min-Max
100 MFC	67.6	61.8	64.7	73.5	70.6	70.6	73.5	61.8	70.6	73.5
200 MFC	70.6	61.8	70.6	73.5	70.6	70.6	67.6	70.6	73.5	76.5
300 MFC	73.5	73.5	76.5	70.6	70.6	70.6	73.5	64.7	64.7	73.5
400 MFC	70.6	73.5	70.6	73.5	70.6	70.6	70.6	67.6	73.5	73.5
500 MFC	73.5	73.5	70.6	73.5	70.6	70.6	64.7	67.6	73.5	70.6

600 MFC	73.5	73.5	70.6	73.5	73.5	70.6	67.6	70.6	73.5	70.6
700 MFC	73.5	76.5	73.5	73.5	73.5	73.5	67.6	67.6	70.6	73.5
800 MFC	73.5	76.5	73.5	73.5	73.5	73.5	70.6	70.6	73.5	73.5
900 MFC	73.5	76.5	73.5	73.5	73.5	76.5	67.6	70.6	70.6	73.5
1000 MFC	73.5	76.5	73.5	73.5	76.5	73.5	70.6	70.6	70.6	73.5

Character 5-grams, no stopwords list

In the cases where the highest accuracy rates were obtained close to the maximum value chosen for the number of most frequent features (800-1000), I went on with the tests up to 2000 most frequent features, but I still got no accuracy rate higher than 79.4%. Therefore, the next step was to look at the misattributed novels for all the combinations of parameters that led to accuracy rates of over 75%.

The most important thing that I noticed was that five novels were never attributed correctly: *Psylla* by Zamfir Arbore, *Iorgu Cosma. Roman original* by Panait Macri, *Căpitan Buzdugan* and *Tâlhăriile banditului Mihale Bunea, născut în comuna Roșu, districtul Ilfov* by N. D. Popescu and *Fuga* by N. G. Rădulescu-Niger. I looked for in-depth information about them in literary history, but the only thing that these texts seemed to have in common was that they all were relatively short novels, just like most of the other novels that were misattributed in the different tests that I made. Two of the files (*Iorgu Cosma. Roman original* and *Fuga*) contained brief metatextual elements, such as a table of contents and some commentaries related to the continuation of the novel in the next issues of the magazine in which it was published. However, the elimination of these elements did not lead to a visible improvement. In the end, together with my mentor, I decided that the best solution for the moment would be to set a limit of a minimum of 20.000 thousand words and to eliminate all the novels shorter than that from the corpus.

After all these subsequent eliminations, I was left with 34 novels in the training set and 21 novels in the test set. From the first test that I made, with words as the unit of measure and Classic Delta as distance, I managed to get a 100% accuracy rate at 900 MFW. The accuracy rate was considerably higher than before for each of the other tests that I ran on the new, shorter corpus. Nevertheless, whereas these results clearly show that Stylo works best on longer Romanian texts, the analysis cannot be considered entirely reliable, especially because of the small set of data on which it was based (only 55 novels). Moreover, two of the five anonymous novels that I had in the initial corpus are shorter than 20.000 thousand words, which means that the most effective combinations of parameters previously found are not even applicable in their case.

Limitations, Conclusions and Future Work

The research that I conducted during my 11-day fellowship at the Trier Center for Digital Humanities brought me to the conclusion that the StyloR package for intuitive users is moderately effective when applied to Romanian literature due to three main limitations.

First of all, the absence of Romanian from the language options of the package means that operations such as pronoun culling and lemmatisation can only be done manually and require both specific lists and minimal to medium programming skills.

Secondly, when used at a basic level, StyloR gave inconsistent results on texts shorter than 20.000 words, which means that the instrument may prove ineffective for a large part of the Romanian emergent novels.

Last but not least, whereas the absence of more training data may have been a particular drawback of my research project, it is also true that there are not many unitary corpora of Romanian literature at this moment and that the digitisation process is particularly hindered by the low quality of the available copies and by the fact that many novels, especially popular novels, have only been published in instalments.

After the discussions I had with my mentor, my plans are directed towards finding some possible ways to compensate for the above-mentioned issues. When it comes to the absence of some options of the package for the Romanian language, my short-term plan is to publish my pronoun list and my code, together with some explanations, in an open-access repository, so that more researchers could use them. After that, since I have this pronoun list and since the automatic tokenization already works well when applied to Romanian, I intend to write to the main developer of the StyloR package and see what other things should be done so that we may have Romanian as a language option on the graphical user interface. Likewise, since the texts in the corpus that I used already have annotated XML versions, I will try to learn how to automatically turn the words into lemmas, so that I may test this kind of data as well. As a sidenote, I should mention that working with lemmatised texts already gave promising results on other Romanic languages, such as France, so I consider it worth trying that with Romanian texts as well.

The other two problems that I have encountered, related to the relative ineffectiveness of the package on short texts and to the lack of training data, are closely interlinked, so there are common solutions that may be tried. At the advice of my mentor, I plan to use sampling in further analysis, which means that I will automatically work with shorter texts, but more data in terms of the number of files. Likewise, I am considering manually splitting the very long texts that I have in the corpus into more samples and placing one of the samples in the training set and the other(s) in the test set, so that there are more authors represented with texts in both the training and the test set.

Since there are a lot of other tests to be done and since I will probably have to readjust the corpus multiple times, I have spent the last days of my fellowship working on the implementation of an algorithm based on loops, that would automatically allow running multiple tests at once. In the end, one of the most important things that I learned during my experience at TCDH was that, in the long run, spending time learning how to do different things automatically, if possible, is a lot better than trying to do everything manually. Although I came to Trier with the aim of testing StyloR at an intuitive level, my conclusion is that even such tests require minimal programming skills when done professionally and systematically.

Bibliography

Corpus:

Pop-Lite: Romanul de consum și subgenurile sale în literatura română din secolul al XIX-lea: editare digitală și analiză de corpus literar asistate de calculator, Available at: https://pop-lite.org/.

Other Resources:

DCRR. (2023). Dicționarul cronologic al romanului românesc de la origini până în 2000. vol. I–II. Cluj-Napoca: Presa Universitară Clujeană.

Eder, M., Rybicki, J., & Kestemont, M. (2016). “Stylometry with R: a package for computational text analysis”. R Journal, 8(1), pp. 107-121. https://doi.org/10.32614/RJ-2016-007. Available at: https://journal.r-project.org/archive/2016/RJ-2016-007/index.html

Eder, M. (2017). “Visualization in stylometry: cluster analysis using networks”. Digital Scholarship in the Humanities, 32(1), pp. 50–64. https://doi.org/10.1093/llc/fqv061
Eder, M. (2018). Authorship verification with the package stylo. Available at: https://computationalstylistics.github.io/docs/imposters
Eder, M., Rybicky, J., and Kestemont, M. (2018). ‘Stylo’: A Package for Stylometric Analyses. Available at: https://github.com/computationalstylistics/stylo_howto/blob/master/stylo_howto.pdf
Evert, S, Proisl, T., Jannidis, F., Reger, I, Pielström S, Schöch C., and Vitt, T. (2017). “Understanding and explaining Delta measures for authorship attribution”. Digital Scholarship in the Humanities, 32(2), pp. ii4–ii16. https://doi.org/10.1093/llc/fqx023
Haverals, W., Geybels, L., and Joosen, V. (2022). “A style for every age: A stylometric inquiry into crosswriters for children, adolescents and adults”. Language and Literature, 31(1), pp. 62-84, https://doi.org/10.1177/09639470211072163
Herrmann, J. B., van Dalen-Oskam, K., and Schöch, C. (2015). “Revisiting Style, a Key Concept in Literary Studies”. Journal of Literary Theory, 9(1), pp. 25-52, https://doi.org/10.1515/jlt-2015-0003
Karsdorp, F., Kestemont, M., and Riddell, A. (2022). Humanities Data Analysis: Case Studies with Python. Princeton: Princeton University Press.
Patras, R., and Pascariu, L. (2024). Profiling-Genre-Signals-in-a-collection-of-Romanian-Novels-with-StyloR [Data set]. Zenodo. https://doi.org/10.5281/zenodo.10890870
Patras, R., and Pascariu, L. (2024, forthcoming). „Profiling Subgenre Signals in a Collection of Romanian Novels with StyloR”. Recent Advances in Digital Humanities. Romance Language Application. Berlin: Peter Lang Verlag.
Schöch, C. (2024). CLS INFRA D3.3 Showcases for the application of CLS methods and tools. Zenodo. https://doi.org/10.5281/zenodo.10912517.
Smith, P., and W. Aldridge (2011). Improving Authorship Attribution: “Optimizing Burrows' Delta Method”, Journal of Quantitative Linguistics, 18:1, 63-88, DOI: 10.1080/09296174.2011.533591.