Exploring pseudonymization based on automated lingustic annotation

Sandra Derbring
MLT Master Programme / Gothenburg University
Language Technology Resources / LT2314
gusderbsa@student.gu.se

Abstract

This paper outlines an implementation and evaluation of adding part of speech tagging and a possibility to exclude text genres to a pseudonymization service. Previous implementation revealed an overgeneration of proper nouns in comparison with manually annotation on the same data. The annotation tool Sparv was used to annotate the data before running the pseudonymization service. Results show that the number of proper nouns decreased for all types of texts when using part of speech annotation. It further shows that the option to exclude certain text genres gives an overall much better performance.

Introduction

When developing tools for natural language processing, researchers need large amounts of data. The access to relevant and suitable data depends on the area of research, but one area where it might be problematic is data that contains personal information. In 2018, the General Data Protection Regulation was introduced as a way of protecting people's right to their own information when providing it to different actors to handle. Of course, not providing one's personal data would be the best way of protecting it, but seeing as this is a practically impossible in reality, another way of solving this problem would be to make sure data is encrypted and anonymized. Article 25 in the EU Commission states that technology that could safeguard data subjects in this respect is recommended to be built into software from start, making the software comply with the requirement of data protection by design and by default. GDPR states further that pseydonymization is one way of reducing the risk of re-identification of a data subject. Most previous literature and research around the subject of anonymization has been done within the medical context and patient data, but following the GDPR legislation, the interest for looking into these matters for the field of NLP has increased. SweLL is an ongoing project at Gothenburg University that aims at building a digital infrastructure for research in

Swedish as a second language. Their data domain is learner corpora - essays by second language learners - where topics often encourage to providing personal details that are good candidates for pseudonymization. To be able to use the data, manual pseudonymization is carried out, but the project also explores and evalutes the potential of using automatizing pseudonymization. The first experiments with this technique has been to implement rules to detect, label and pseudonymize information. Given language resources for names, geographic data, professions and similar categories are used to detect relevant strings in the data and also for replacing the information detected as sensitive. The rules are for example based on regular expression, contextual triggers and frequency. The resulting code has been integrated into the SweLL annotation tool, SVALA, so that the results can be tested and inspected in a user-friendly way. The results of the first experiments with the implemented techniques show that the pseudonymizer captures personal information quite well within texts that are narrative, argumentative and instructional compared to manual annotation. However, compared to manual annotation for texts that are evaluative or investigative, such as news articles or book reviews, where names and places are commonly occuring but don't reveal sensitive data about the essay author, there is an overgeneration of pseudoymization in the automatic implementation. Results also show that the use of capitalization of words leads to overgeneration in the automatic service, since capitalized common words are mixed up with personal names and cities. The goal of this service is to collect essays on the fly - that is, to be able to automatically pseudonymize texts in a accurate way with as little errors as possible compared to a manual assessment. The results of these first experiments suggest that there are different ways to improve the implementation to get closer to the end goal. Improving the algorithm to do this is the basis for the work presented in this paper. In section 2, methods and materials will be discussed. Section 3 will show results from experiments compared to those already performed with the original implementation and with manual annotation and section 4 will discuss those results and reflect on how the implementations affected the outcome. Finally, section 5 will conclude the work and propose directions for further work.

Method and materials

The experiments suggest several directions for future work and improvement of the services. One of those is to apply a part of speech tagging to the text to disambiguate proper nouns from other parts of speech such as prepositions. Another is to implement a topic selector that excludes certain texts from being pseudonymized at all. In this section, the process of implementing and evaluating those alterations will be presented.

The data

The data used for testing the implementation is a set of 85 manually pseudonymized essays written by second language learners. The essays can be classified into one of three levels; beginner, intermediate and advanced lingustic competence, and into one of five genres; narrative, argumentative, instructive, investigative and evaluative. Statistics from the manual pseudonymization shows that essays on the lower levels need pseudonymization to a higher extent. Additionally, essays in the narrative and argumentative genres include content that is prone to a greater need for pseudonymization. Topics within these genres include topics such as “About your accomodation” and “Describe a place where you live”. In contrast, the other genres’ topics are “Book review” or “Discuss work moral”, where it’s apparent that the level of personal information is lower.

The original implementation

The implementation of the original pseudonymization is done in python. It is split into three steps; detection of personal segments, labeling of categories and pseudonymizing. For both detection and replacement, data resources are necessary. For this project, open available official statistical agencies have been used to collect geonames, personal names, companies, street names, languages, professions and universities and processed to be usable for this purpose. In the detection step, the strings in the text are matched against different rules and expressions in order to be assigned a category. There are three groups; hard replacement (regular patterns or numerical expressions), placeholder (geographical information, personal names and institutions) and sensitive (professions, family relations, religious, ethnical, sexual and political information). In this ongoing work, there are various discussions around particularly the more delicate categories in the sensitive group and how to best deal with them. The labeling step attempts to make an additional assessment on strings that are ambigous and may belong to several categories and assign it to the most likely. Finally, the pseudonymization step is where operations are made on the strings detected and labeled into the hard replacement group or the placeholder group. For the numerical or regular pattern strings in the hard replacement group, the replacement with similar numbers or pattern is straightforward. Strings categorized into the placeholder group are replaced with strings from the previous mentioned resources and have to be matched in gender, geographical areas and morphological case.

Hypothesis 1: Introducing POS

One of the discoveries from previous experiments was that one of the causes for diskrepancy between the manual annotation and the automatic annotation was that the implementation mistakenly matched common function words to be proper nouns. This resulted in an overgeneration of mainly personal or geographical names - for example, the possessive pronoun ’hans’ (’his’) matched in the name database for the male first name Hans. Following this, the hypothesis was that a part of speech tagging of the data would help reduce these ambiguities. To test this hypothesis, an annotation step was added to the original implementation. In this preprocessing step, also implemented in Python, the annotation tool Sparv, developed by Språkbanken (ref), was used. Sparv has a web user interface where plain texts can be inserted and that returns an annotated version of the text. It also has an API which makes it possible to use from other programs. The annotation step starts with tokenizing the data file it gets as input. For each sentence, it then uses Sparv's API to get the annotations for the words. The Python package urllib was used to request, parse and read the data from the website. [Example?]. The annotated data that is returned from Sparv is in XML format. Therefore, an XML parse has to performed on the string to extract the right information. The Python package XML Element Tree was used to iterate over each word element and within it, iterate over the attributes to find the 'pos' value. The word and the part of speech tag is then concatenated into a string in this manner: word+'//'+pos. The double slash sign is used to easily be able to perform detecting and splitting operations on the string. This annotation step finally returns a list of those annotated sentences. [Example?] In the original implementation, the plain data files was sent as the only argument to the identification step. Now, this list of annotated sentences is sent as a second argument. Instead of using only this as the input, and thus having to implement a funtion for splitting each word from its tag, the annotated list is treated as an additional source. The identification implementation iterates through the regular plain input as before. Each time a word is matched against any of the data sources and identified as either a country, city or proper name (first name or surname), the sentence's index in the text and the word's index in the sentence is extracted. These indices are sent to a specific function where the corresponding word in the corresponding sentence is extracted from the annotated data. If the part of speech tied to this word is 'PM', which tells us it is a proper name, the function returns true. This boolean value then works as a condition for if the word should be processed further as an identified geographical or proper name and, in the end, be labeled and/or pseudonymized. The hypothesis is that this will filter out words that otherwise get falsely labelled.

Hypothesis 2: Introducting topic selection

The other main reason to why the original implementation produced an overgeneration of proper nouns was that all genres are treated the same, even if topics such as argumentative articles or reviews of books or movies seldom reveals many personal details about the author. Typical for these kinds of texts is instead names and places relating to authors, book characters or other references, which do not need to be pseudonymized. This analysis resulted in the hypothesis that the possibility of excepting certain genres could help reducing the number of matches. To test this hypothesis, a simple switcher was implemented as a very first condition. In this version, it is implemented as a boolean argument that the user specifies (on or off). Each file name in the sample contains what kind of genre it belongs to. This implementation therefore contains a dictionary that maps between the genre codes and the names for the genres as well as a list of genres that should be excepted. If the user has chosen the exception setting to 'on', the script checks what genre the given file belongs to and if that genre should be excepted. If the answer if yes, the file is not processed at all. If the setting is 'off' or if the genre is not to be excepted, the file is processed as usual. [Example on whole process as a flow chart, see Elena's article and add both your parts] [Example of a file name]

The pre-processing and evaluation setup

To be able to evaluate the implementation on a number of files, the sample data of 85 files were used. This data was given in a json format and consisted of three parts; the source text, the pseduonymized texts and arcs between them. To be able to run the scripts, the data was needed in plain format. A script to extract the words from the json format was created, which returned a plain text. The evaluation is setup to be able to test the different functions and return statistics. It's possible to, as a user, to run the scripts with or without part of speech annotation and with or without topic selection. Additionally, it is possible to limit the number of words in the file, which is useful while iteratively testing features and thus avoid long runtimes. The script accepts a folder of files as an input argument. It then iterates over the files in that folder and run them one by one through the different scripts. If topic selection is turned on, it checks the file's genre as described above, and processes depending on the outcome. If part of speech tagging is turned off, the script starts the original implementation. The returned labelled and/or pseudonymized data from the identification script is then sent to an evaluation script, which counts the number of words for each label. Each label and its count is then assigned to the genre the current file belongs to. If the file belongs to several genres, this data is assigned to all of them. When all files have been processed, statistics about the number of hits for each label for each genre is created. As a further step to analyse the results for the experiments for this report, statistics about which words are labelled as country, city or names are shown to be able to detect consequential errors.

Results

Original results

Figure 1 shows the number of automatic versus manual labels per genre and label from the original code. It's the results from this table that is the basis of this project. The table shows what has been discussed previously - that the detection, labeling and pseudonymization works well for the texts in the narrative, argumentative and instructional genres but overgenerate for investigative and evaluative texts. As can be seen in the table, some labels are only used by the manual annotation (e.g. area) and some labels are only used by the automatic annotation (e.g. island). The labels were initially defined to be used for manual annotation. The reason for the first diskrepancy is that there are no digital sources to gather information about area and thus it was left out of the implementation. The reason for the second is that a digital source for islands was found and thought to be of good use to improve the annotation and therefore was added.

Picture of the original table with results

Tag	**Man**	**Auto**	**Man**	**Auto**	**Man**	**Auto**	**Man**	**Auto**	**Man**	**Auto**	**Man**	**Auto**
age_digit	10	8	0	0	0	0	0	0	0	0	10	8
area	14	0	8	0	0	0	0	0	0	0	22	0
city	13	18	0	7	0	1	0	12	0	44	13	82
city_swe	25	31	10	9	4	4	0	1	0	0	39	45
country	30	16	17	5	0	0	0	0	0	0	47	21
date_digit	0	3	1	5	1	0	0	0	0	0	2	8
day	2	2	4	0	0	0	0	0	0	0	6	2
edu	0	0	1	0	0	0	0	0	0	0	1	0
email	0	0	1	1	0	0	0	0	0	0	1	1
extra	1	0	1	0	0	0	0	0	2	0	4	0
1:female	18	16	7	5	2	3	0	50	0	45	27	119
1:male	4	10	13	8	6	2	0	48	1	73	24	141
1:neutral	10	12	2	21	2	7	0	55	2	17	16	112
geo	2	0	0	0	0	0	0	0	0	0	2	0
month-dgt	0	0	1	0	0	0	0	0	0	0	1	0
month-str	2	2	3	2	0	0	0	0	0	0	5	4
other_nr	0	0	1	0	0	0	0	0	0	0	1	0
phone_nr	0	0	3	3	0	0	0	0	0	0	3	3
place	7	0	6	0	0	0	0	0	0	0	13	0
prof	0	11	1	2	0	0	0	12	0	3	1	28
region	3	0	3	0	0	0	0	0	0	0	6	0
sensitive	2	34	0	14	0	1	0	7	0	53	2	109
street_nr	1	4	4	7	0	0	0	10	0	2	5	23
surname	2	9	6	7	1	3	0	81	3	15	11	115
transport	5	0	10	0	0	0	0	0	0	0	15	0
year	3	3	4	5	0	0	0	18	0	6	7	32
zip-code	0	0	3	0	0	0	0	0	0	0	3	0
TOTAL	**154**	**179**	**110**	**101**	**17**	**21**	**0**	**467**	**8**	**276**	**289**	**1044**

Baseline results

As a first step in the evaluation, the original script was run on the available data to get a baseline for the remaining experiments. The expectation was to get a complete match with the results previously reported from the project. However, this was not the case. There are labels present in the original results that are not produced in this run, and vice versa, there are some labels produced in this run that do not exist previously. The reason for those diskrepancies is that the script is constantly evolving during the project time and it has been tweaked and adapted in correlation to the project group's discussion about definition, inclusion and exclusion for various labels. To run the original code was thus an important thing to do to be able to get a reliable baseline to compare the further experiments to.

Results from the part of speech annotation

Figure 3 shows the results from the part of speech annotation per genre and label as well as from the original code. For clarity, only labels affected are included in the table. For all other labels, the numbers remain the same between the baseline and the addition of part of speech annotation. The manual annotation is included in the table as well to give a view of the golden number. As the labels have changed during iterations, the exact number of manual annotations might not be totally correct given the labels in this experiment but nevertheless they give a hint of the correct extent of matches. We can see that the in general, inserting of part of speech annotation decreases the matches for all of those labels, and generally comes closer to the number of manual annotations. At times, the automated approach even matches fewer names than the manual annotation as a result of this intervention. Looking at the total number of matches for each genre, using part of speech annotation gets us very close to the manual annotation for the narrative, argumentative and instructional genres. For the investigative and evaluative genres, however, the diskrepancy between the manual and automatically annotated numbers is very high, and the decrease that the part of speech tagging gives does not at all come close.

city	17	6	13	4	2		1			12	2		39	30	
country	6	3	30	3	2	17				135	135		18	16	
firstname__female	11	10	18	4	3	7	3	3	2	49	35		37	32	
firstname__male	10	3	4	7	3	13	2		6	42	29		68	55	1
firstname__unknown	15	4	10	17	7	2	6	1	2	50	19		17	6	2
surname	11		2	7	1	6	3		1	66	63		13		
TOTAL	**70**	**26**	**77**	**42**	**18**	**45**	**15**	**4**	**11**	**354**	**283**	**0**	**192**	**139**	**3**

Results from both part of speech annotation and topic selection

Figure 4 shows an overview over results from different combinations of setups for the relevant labels. The manual annotation is included, but just as before, it's not an exact comparision. We can see that while the part of speech annotation does have an decreasing effect on almost every genre, what really gives a difference in the total number of annotations is the exclusion of text from the investigative and the evaluative genres. In fact, the numbers seem to suggest that using only the topic switcher, without the part of speech annotation intervention, gives the closest match to the manual annotation number. Using both decreases the number far below the manual assessment.

			Manual		
	Without topic	**With topic**	**Without topic**	**With topic**	
Tag					
city	73	22	40	8	13
country	162	9	156	5	47
firstname__female	104	18	83	16	27
firstname__male	129	19	90	6	24
firstname__unknown	105	38	37	12	16
surname	100	21	64	1	11
TOTAL	**673**	**127**	**470**	**48**	**138**

Discussion

In this project, the research question was to evaluate whether an implementation of part of speech annotation and a topic switcher would enhance the result of automatic pseudonymization. The hypothesis was that both those new implementations would decrease the overgeneration of proper nouns. Results from experiments from these approaches show that the hypothesis holds true. Both approaches decrease the detection and pseudonymization of proper nouns. The part of speech annotation gives an effect on each separate genre. The topic switcher does not give an effect on those genres still included, but of course decreases the number of proper nouns detected with 100% for the genres excluded. With those approaches combined, the number of proper nouns found is actually lower than the manual annotation. This result is interesting and should be an object for further experiments and investigation. Does that mean that the part of speech annotation is, in practice, unnecessary and that the topic exclusion alone would suffice to use in practical implementation? To be able to see what causes these results, it would be desirable to do a closer analysis into what words are matched and which are not, to see if there is something that could be implemented differently to enhance the results. It will be important to not just look at the number of words categorised to a certain label, but also which ones. We need to make sure two things; that the words categorised are correctly classified (i.e. is a proper noun of that category in the context) and that there are not an abundance of words that are missed (i.e. false negatives). A first step towards this has already been implemented in the evaluation and those results show that the words found belong to the correct category (country, city or personal name). It would be desirable to develop this check by also look at the words being sorted out by the part of speech taggning, and by cross-matching against the arcs in the service. Of course, it would be also ideal to have a better match between the manual annotation and the automatic one - that the labels are the same, so that it would be easier to interpret comparisons and catch systematic errors in the algorithm. A built-in problem in the detection and categorisation of words is that a proper name can belong to several categories at once - the same name can be both a city and a first name (such as Elena), or both a male and a female name (such as Kim), or both a first name and a surname (such as William). The larger the resources are, mixed with names from all over the world, the risk of cross-matchings increases. In the original implementation - which is not changed by the intervention of adding part of speech tagging - the algorithm makes a decision as to which category is most likely that the name belongs to. This logic would be interesting to inspect further, for example in conjunction with the inspection of which names are included or excluded as mentioned above. Another cause that would be interesting to inspect further is how multi-word expressions are handled. In the part of speech taggning by Sparv, not all words, such as street names and countries consisting of several words (Gamla Anneforsvägen, Papua Nya Guinea) are annotated as one entity, but instead of several (where some get a proper noun tag and some are tagged as for example an adjective). In the original implementation, the data sources are iterated over in order and once a word matches a source in the database, it does not look further. For example, in the sentence "I grew up in Papua Nya Guinea", the algorithm would tag "Guinea" as the city, but not "Papua Nya Guinea", as "Guinea" is in the database and is found before "Papua" given the alphabetical order. A similar example, "New York" and "York" would not get the same outcome, since "New York" would be found first. So the sentence "I grew up in Papua Nya Guinea" could be pseudonymized into "I grew up in Papua Nya England". Things like this might therefore impact the pseudonymization and it would be interesting to see if there could be improvements made to take account for them. Something that also plays a part here is which language the geographical names are written in. I would think that it wouldn't be uncommon that in texts written by people with Swedish as their second language, cities and countries could be expressed in their English form (or in another language) - in which case they would not be detected by the database where the geo names are in Swedish.

Conclusions and future work

This paper has examined the hypothesis that part of speech tagging and topic selection would improve the existing pseudonymization service developed in the SweLL project, whose previous results show an overgeneration of proper nouns compared to manual annotation of the same data. This project has added those two enhancements to the original implementation. The part of speech tagging is done with the help of the annotation tool Sparv, developed by Språkbanken and the topic selection is a simple user controlled condition that excludes text from certain genres. Results from experiments from these approaches show that both decreases the number of detected proper nouns, but that excluding the problematic genres is most effective in total. The discussion concludes that a further analysis of which words are included and excluded with these approaches is needed to fine-tune the implementation. The question of the influences of the languages used for geonames, the handling of multi-word expressions and the algorithmic selection of which category a name belongs to is also discussed. For all of those, further work could be done to analyse and/or improve the outcome. In addition to those, there are further work do be done concerning the current implementation of the part of speech tagging and the topic selection, work described within this paper. In this first version, only countries, cities and proper names are being checked for a proper noun tag, since those were the areas most concerned with overgeneration. In a future version, additional categories could be candidates for benefitting from this as well, such as islands, professions or universities. The topic selection switcher could be implemented in different ways depending on how it fits to be integrated into the rest of another implementation. Currently, the user must specify this at the command line when running the file. It could instead be a dialogue question when the script is started, and also give the user a possibility to choose which genres to exclude. If this approach is to be implemented in the service that has a graphical user interface, it should be added as a checkbox or similar for the user to be able to control. Further on, it's desirable to perhaps to do a real topic detection analysis instead of this approach, which takes their hints from the filenames. This works for this sample data used currently, but won't last in the long run on any given files that someone might want to use this service for.

Acknowledgments

I would like to thank Elena Volodina and Yousuf Ali Mohammed for their encouragement and all help providing the original code, resources and being available for questions and thoughts throughout the process.