Gender Discriminatory Language Identification with an Hybrid Algorithm based on Syntactic Rules and Machine Learning

Valerio Bellandi[1], Stefano Siccardi[1]

[1] Computer Science Department, Università Degli Studi di Milano, Via Celoria 18 Milano (MI) Italy

Abstract

In the last years, gender discrimination in textual documents has emerged as an open problem and is undergoing analysis. The difficulty of identifying sentences in which this discrimination is present is linked to the context used and the formalisms adopted. This work describes an exploratory activity linked to the context of regulations and official documents of Italian public administrations. A hybrid algorithm based on syntactic rules and machine learning is therefore proposed, capable of identifying a specific subset of possible gender discrimination.

Keywords

Gender Discrimination, Syntactic Rules, Entities Extraction

1. Introduction

Discriminatory attitudes against minorities or gender related have been reported in many areas; often they are conveyed by language in the form of open "hate speech" or in more subtle forms, for instance associating the discriminated group to specific social roles or professions. Several social networks, like Facebook and Twitter, define and ban hate speech (see [4]). Nevertheless, [9] analyzing more than 2 millions tweets in a 7 months period, found that women (at the first place), immigrants, gay and lesbian persons, Muslims, Jews and disabled persons were addressed by more than 100 thousands of hateful tweets. Another study ([7]) reports that around 10% of social media users reports being victimized by online hate speech. On the other hand, several institutions have approved guidelines to promote the usage of an inclusive language in their official documents, that is a language that does not carry any explicit or implicit difference between genders. For instance, we quote the European Parliament (see [8]) and the University of Milan (see [10]). This work aims at helping the detection of non inclusive language usage to facilitate their correction.

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy

valerio.bellandi@unimi.it (V. Bellandi); stefano.siccardi@unimi.it (. S. Siccardi)

0000-0003-4473-6258 (V. Bellandi); 0000-0002-6477-3876 (. S. Siccardi)

2. Related work

Natural Language Processing techniques to mitigate gender bias have been reviewed by [15]. They start from the observation that NLP systems containing bias in training data, resources, pretrained models (e.g. word embeddings), and algorithms can produce gender biased predictions and sometimes even amplify biases present in the training sets. An example, driven from Machine Translation is that translating "He is a nurse. She is a doctor." to Hungarian and back to English results in "She is a nurse. He is a doctor." The main contributions of the paper are related to NLP itself, more than to user composed documents. In the same spirit, [2] quantify the degree to which gender bias differs with the corpora used for training. They look especially at the impact of starting with a pre-trained model and fine-tuning with additional data. A different strand of study uses sentiment analysis to try to detect gender of writers in blogs or social media, for instance to establish a detailed health policy specialized into patient segments ([12]), or aims at detecting gender and age of writers in order to improve sentiment analysis ([16]). [1] tested the use of tools to simplify the task of information selection from collections of documents used by qualitative researchers in order to analyze discrimination. They compare some methods and find that relevant words can be efficiently found, but results heavily depend on the quality of pre processing. A tool to check German texts for gender discriminatory formulations has been described in [5]. It is based on rules to detect nouns used to denote male persons and then to filter out correct cases, for instance when the noun refers to a specific male person (as a proper noun is found nearby). In non correct cases the user is prompted with suitable messages and hints. The way this tool works is basically the same of our rule based model. More recently, Microsoft Word text editor started offering a tool to check for non inclusive language, that, when activated, prompts some hints during documents input. However useful, it is presently not configurable and does not cover a large number of cases. Our method relies on Named Entity Recognition capabilities of NLP software to extend the set of entities to check; a complete review of the topic is however out of the scope of the present work and we will limit ourselves to describe some specific points. [11] examined 200 English and 200 Spanish tweets containing hate speech against women and immigrants using a model based on Spacy ([14]). Words were divided into 27 categories according to the type of negative contents; an accuracy of 84% for English and 62% for Spanish in identifying hate speech is reported. Bert has been used in [13] to analyze 16,000 tweets, containing 1972 tagged as racist and 3383 as sexist. A goal of the study was to reduce false positives to detect hate speech without undermining the freedom of expression. The maximum achieved specificity, was 83.03. Another study, [3], compared four algorithms including Spacy, Bert and two monolingual algorithms (Flair and camemBERT) to find entities in a set of 500 legal French cases, where a pool of experts annotated the cases with 60,000 entity quotations. The monolingual tools reached the best precision and recall.

3. Method

The proposed methods consists of two interleaved pipelines, one is the "production" tool for the detection of discriminatory language in official documents, the other is used to expand the tables used by the system, so that more and more cases can be detected. Referring to fig.

1, the first pipeline, labelled 1, is fed with documents. A detection software, using rules and tables with words used as "seed entities", produces a tagged version of documents, to highlight potentially discriminatory points. Each word included in the tables and not fitting one of the required rules is tagged as an entity (in the sense of NER of NLP terminology), with a suitable error tag. A user checks the tags, fixes the document and resubmit it, until a satisfactory result is obtained. Examples of tables are: entities to check, like: (man: woman), (men: women); list of male proper names, and so on. Examples of rules are:

1. Base rules, evaluating True or False: R1 = (word is in table of entities); R2 = (one of neighbors of word = table of entities[word])
2. Compound rules, evaluating a Tag or None: if R1 and R2 then assign Tagxxx to word

The second pipeline is in turn divided into two branches. The branch labelled 2 builds a model and the branch labelled 3 uses it to find new entities. In branch 2, the documents are processed by a modified version of the detection software, that shares the same rules and tables. However, instead of creating tagged versions of the documents, it creates a set of annotations. These are single sentences, taken from the documents, with an occurrence of an entity, tagged with the proper error, in the format required by the training program of the chosen NLP tool. Moreover, whenever a "wrong" occurrence is found, all the "correct" version are added, as separate annotations with the proper tags. The annotation are then used to train a NER model. We used Spacy, but in principle any NLP system with trainable NER capabilities can be employed. In branch 3, the model is fed with the documents and produces a new set of tagged ones. Tagged words will not coincide with those found in pipeline 1; often we found more errors compared to the rule based version. However new entities, not in the set of the seed ones, are in general found. After a user's review, the approved entities are added to the tables used in pipeline 1. The role of the user is important, because the system often finds entities that are not of interest or even completely wrong. This closes the loop, extending the capabilities of the "production" rule system. Pipeline 2 can be run with the extended tables, until a satisfactory set of terms is obtained.

4. Experiments and Results

A selection of official documents of the Milan University, chosen among general, departmental, security / privacy and staff regulations has been used to search for two basic types of "problems": *i)* sentences containing only the male form of a noun having a different female form, e.g. "uomo – donna" (man – woman) *ii)* sentences containing nouns having the same male and female form, without any other grammatical element to stress reference to both genders, e.g. "il docente" instead of "il/la docente" (the teacher, no English analogous article form) Whenever one of the above was found, the following annotations were created:

- the original sentence,
- the sentence with both forms of the noun for cases 1 above,
- the sentence with the unique form of the noun and both articles for case 2 above,
- the sentence with the male form of the noun and male article, followed by a randomly, chosen proper male name,

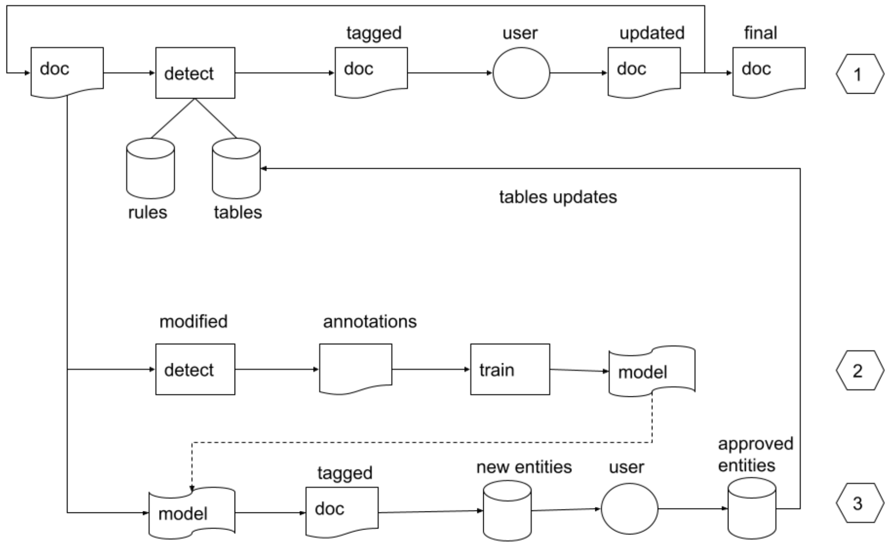

Figure 1: The pipelines used by the method.

- the sentence with only the female form of the noun or the female article.

We used these annotations to train the model to recognize the possible cases, assigning a label to each one.

4.1. Training with a rich seed entity set

We used a set containing 23 entities typically found in University documents, like teacher, student, researcher and so on and trained two different models, the first using 4683 annotations the second one using 8272. After training, both models have been used to analyse the same set of documents. In order to evaluate the accuracy, as a first approximation we considered the rule based detector 100% correct for the seed entities and implemented a semi automatic procedure to check detection errors (false positives). We found that the rule based model found a total of 1846 errors, the first model 2337 including 634 false positives, so that accuracy is 72.9%; it missed 143 errors found by the rule based model, that is the 7.7%. The second model found 2316 errors including 503 false positives, with accuracy 78.3%, missing 33 errors (1.8%) found by the rule based model. The first model was able to detect 23 new correct entities (e.g. "referent", "warranter/sponsor"), the second one just 16. In other terms, a larger training set reduces errors, but also the number of new entities found. It must be noted that the rule based model was actually not 100% correct, a circumstance that may have had a negative impact on Spacy's training. A manual check performed on $\approx 15\%$ of the documents showed ≈ 4.47 false positives. For example the term "componente" (component) may indicate a person in a sentence like "un componente dello staff..." (a component of the staff) or an abstract entity in the sentence "la componente studentesca della popolazione" (the student component of the population).

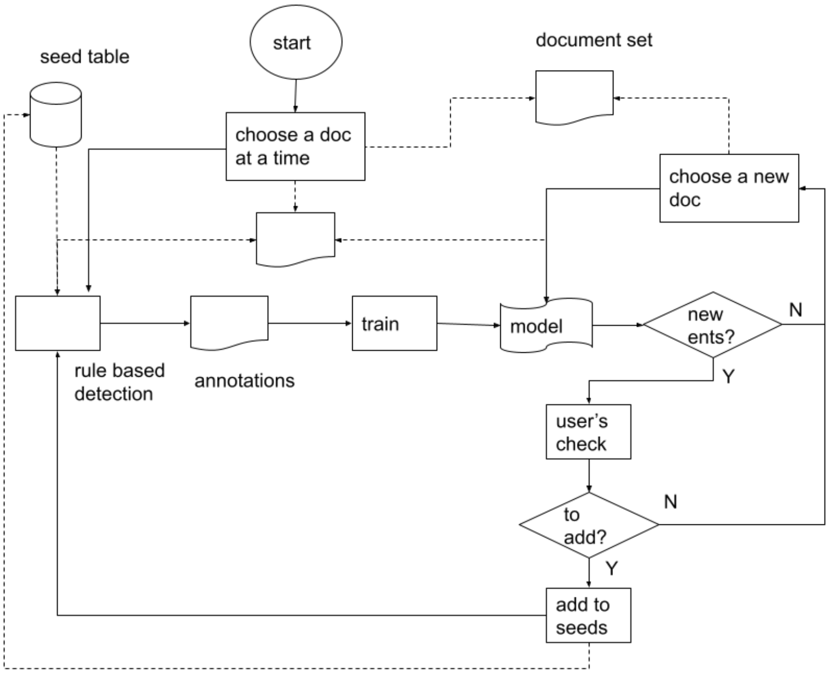

Figure 2: The workflow of incremental training.

4.2. Training with an incremental seed entity set

Experiments were performed to check the ability of build a rich set of entities starting with just a few ones. The workflow is shown in fig. 2: we started with a table consisting of a few seed entities and used the rule based model to create annotations from a document. They were used to train a model, that was run on the same document. If some new entities were found, they were manually checked and, if correct, added to the entity table; then new annotations were created and the model was trained again. If no new entities were found, a new document was used without retraining the model. In the picture, continuous arrows denote the control flow, dashed ones the data used at each step.

Results are summarized in table 1.

Two experiments were run, the first starting with 7 seed entities (the first 3 columns in the table 1), the second with 3 (the last 3 columns 1). Each row shows the number of new entities found if any, the number of corresponding annotations and the time in minutes needed to retrain the model. Experiments were run on a "small" system, that is a commercial PC with an i7 Intel processor, 6 cores, 2.7 GHz, 16 Gb Ram. We can summarize that:

- the first experiment stopped finding new entities after the 22nd
- 10 entities found by the first and 6 found by the second model were elements of the original 23 seed entity set chosen by the user for the experiments in section 4.1

New ents	Annotations	Train time		New ents	Annotations	Train time
7	66	1		3	3	0.2
0	—	—		0	—	—
1	1069	3		2	1894	11
2	1107	5		0	—	—
0	—	—		0	—	—
0	—	—		2	1928	9
1	1137	5		0	—	—
1	1339	13		0	—	—
2	1977	16		1	1933	19
0	—	—		1	1933	12
2	1983	16		1	1971	10
2	2019	8		3	1977	11
2	2168	8		2	2054	13
2	2173	7		3	2676	17
0	—	—		3	2818	20
0	—	—		0	—	—
0	—	—		0	—	—
0	—	—		2	2861	18
22				23		

Table 1
Results of incremental training.

- some “popular” entities sharply increase the number of annotations
- the training time roughly increases linearly with the number of annotations, even if it does not depend only on it (see fig. 3)

5. Conclusion and future work

We showed that a combination of a rule based model with a trainable one is a promising way to get an accurate and extensible tool to detect non inclusive language. We applied the method to official documents of an Institution as a first test area, but we think that it can be applied to wider types of documents and discrimination languages. In the future, we plan to consider, for instance, textbooks and blog or newspaper articles. These imply to manage some more subtle types of discriminatory language, related for instance to bad sentiments and stereotypes. Therefore, we plan to perform some technical enhancements, such as: *i)* the rule engine and rule set will be expanded to handle more complex cases and avoid the small percentage of errors we found in the present work *ii)* we will compare performances of several Natural Language Processing tool, instead of using just one and *iii)* we will include methods of the Sentiment Analysis area.

Acknowledgements

The authors wish to thank their students Francesca Aru, Giovanni Lafiandra and Giulia Pagani for their helpful work, especially during the experimental phase. This work was partly supported by Università degli Studi di Milano under the program “Piano di sostegno alla ricerca”.

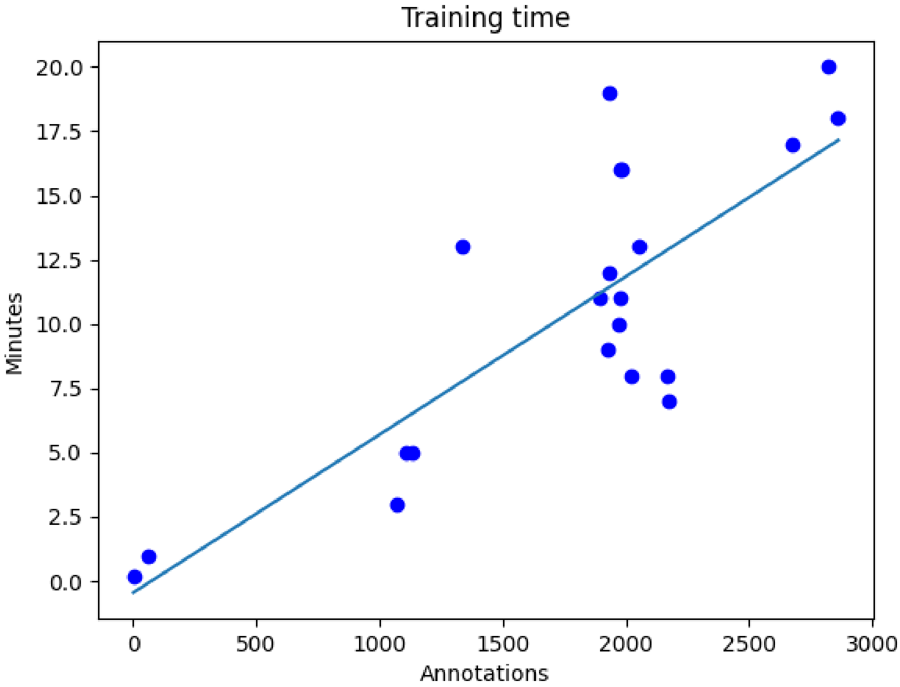

Figure 3: The dependence of training time on annotation number.

References

[1] Alatrista-Salas, H. and Hidalgo-Leon, P. and Nunez-del-Prado, M. Documents Retrieval for Qualitative Research: Gender Discrimination Analysis, 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), 2018, pp. 1-6, doi: 10.1109/LA-CCI.2018.8625211.

[2] Babaeianjelodar, M. and Lorenz, S. and Gordon, J. and Matthews, J. and Freitag, E. ,Quantifying Gender Bias in Different Corpora. Companion Proceedings of the Web Conference 2020. Association for Computing Machinery, New York, NY, USA, 2020. DOI:https://doi.org/10.1145/3366424.3383559

[3] Benesty, M., NER algo benchmark: spaCy, Flair, m-BERT and camemBERT on anonymizing French commercial legal cases, https://towardsdatascience.com/benchmark-ner-algorithm-d4ab01b2d4c3, Last accessed 22 Feb 2022

[4] Bortone, R. and Cerquozzi, F., L'hate speech al tempo di Internet, Aggiornamenti sociali, vol. 818, 2017

[5] Carl, M. and Garnier, S. and Haller, J. and Altmayer, A. and Miemietz, B. Controlling gender equality with shallow NLP techniques. In Proceedings of the 20th international conference on Computational Linguistics (COLING '04). Association for Computational Linguistics, USA, 820–es. 2004. DOI:https://doi.org/10.3115/1220355.1220473

[6] Devlin, J. and Chang, M.-W. and Lee, K. and Toutanova, K., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), url https://aclanthology.org/N19-1423, doi:10.18653/v1/N19-1423, 2019

[7] Döring, N. and Mohseni, M. R., Gendered hate speech in YouTube and YouNow comments: Results of two content analyses, SCM Studies in Communication and Media, vol. 9, n. 1, 2020

[8] Gender neutral language in the European Parliament, https://www.europarl.europa.eu/cmsdata/151780/GNL_Guidelines_EN.pdf. Last accessed 28 Feb 2022

[9] Lingiardi, V. and Carone, N. and Semeraro, G. and Musto, C. and D'Amico, M. and Brena, S., Mapping Twitter hate speech towards social and sexual minorities: a lexicon-based approach to semantic content analysis, Behaviour & Information Technology, vol. 39, n. 7, 2020

[10] Linee guida per l'adozione della parita di genere nei testi amministrativi e nella comunicazione istituzionale dell'Universita degli Studi di Milano, (in Italian), https://www.unimi.it/sites/default/files/regolamenti/Lineeguidalinguaggiodigenere_2020_UniversitádegliStudidiMilano.pdf. Last accessed 28 Feb 2022

[11] Lai, M. and Stranisci, Marco A. and Bosco, C. and Damiano, R. and Patti, V. HaMor at the Profiling Hate Speech Spreaders on Twitter, Working Notes of CLEF 2021-Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, vol. 2936, pp. 2047–2055, 2021

[12] Park, S. and Woo, J. Gender Classification Using Sentiment Analysis and Deep Learning in a Health Web Forum. Appl. Sci. 2019, 9, 1249. https://doi.org/10.3390/app9061249

[13] Gaurav Rajput and Narinder Singh punn and Sanjay Kumar Sonbhadra and Sonali Agarwal, Hate speech detection using static BERT embeddings, arxiv:2106.15537, 2021

[14] Spacy Homepage, https://spacy.io/. Last accessed 21 Feb 2022

[15] Tony Sun, T. and Gaut, A. and Tang, S. and Huang, Y. and ElSherief, M. and Zhao, J. and Mirza, D. and Belding, E. and Chang, K-W. and Yang Wang, W., Mitigating Gender Bias in Natural Language Processing: Literature Review, arxiv 1906.08976, 2019

[16] Volkova, S. and Wilson, T. and Yarowsky, D., Exploring Demographic Language Variations to Improve Multilingual Sentiment Analysis in Social Media, url=http://aclweb.org/anthology/D/D13/D13-1187.pdf, EMNLP. 2013