# Detecting Offensive Language in Romanian Social Media

**Article** *in* Procedia Computer Science · October 2022

DOI: 10.1016/j.procs.2022.09.346

Diana Trandabat
Universitatea Alexandru Ioan Cuza
**71** PUBLICATIONS **300** CITATIONS

SEE PROFILE

Daniela Gîfu
Romanian Academy
**187** PUBLICATIONS **437** CITATIONS

SEE PROFILE

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2022) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

# Detecting Offensive Language in Romanian Social Media

Diana Trandabăț[a]*, Daniela Gifu[a,b] , Pleșescu Adrian[a]

[a]*Faculty of Computer Science, "Alexandru Ioan Cuza" University, General Berthelot, 16, 700483, Iasi, Romania*
[b]*Institute of Computer Science, Romanian Academy - Iasi branch, Bulevardul Carol I, 8, 700505, Romania*

## Abstract

Due to an exponential increase in the use of Internet by persons from different countries and educational backgrounds, the offensive online language detection has become a significant task facing natural language processing. Considering the major negative impact of this type of content in the case of youngers, detecting online toxic language to protect users' online safety becomes an urgent issue. The project has two main goals: (1) developing an annotated corpus of offensive content for Romanian language and (2) testing various machine learning algorithms to identify a best approach. The proposed methods achieve results with a few percentages more than the accuracy of the current SoTA.

*Keywords:* offensive language; social media; machine learning; toxic comments.

## 1. Introduction

Given the volatility of online content [1], users are spending increasing amounts of time on various social networking sites to get informed and gossip with peers. This kind of interaction brings a considerable amount of offensive content, encouraged by the physical distance. Natural language processing (NLP) has started to use its technologies to identify such unwanted contexts, referred to as offensive, aggressive or toxic language. Romanian is a less resourced language, that is why we continued the challenge of detecting offensive language [2] with an emphasis on Romanian.

* Corresponding author. Tel.: +40 232 201771
*E-mail address:* dtrandabat@info.uaic.ro

Automatic offensive language recognition in social media aims to help administrators monitor and filter content to ease a process which is time and resources (humans) consuming. But social media administrators are not the only target group for such a system. Recently, government entities are joining the fight against online harassment, especially in the case of children and teenagers.

A recent article [3] discusses the differences of training an NLP model on 2 online encyclopedias: Wikipedia and its Chinese equivalent, Baidu Baike. The researchers found proof of what they initially believed, namely that censorship influences the classification of words based on their political meaning: for example, "democracy" is more related to "chaos" then it is to "stability". As a consequence, a news headline analyzer built on top of this censored corpus had a bias in considering ideas as propaganda, especially in the case of non-patriots.

The research question this paper intends to answer is *How much machine learning (ML) can help to recognize offensive language in order to avoid the online toxicity?* We answer it by proposing a system able to identify whether a comment is offensive or not, based on various ML algorithms.

## 2. Related Work

Toxic communication has invaded social media content [4]. It is also the reason why in recent years we see evidence of increasing number of international research events such as SEMEVAL-2019, ALW 2017÷2021 (*Workshop on Abusive Language Online*) or TRAC 2019÷2020 (*TRolling, Aggression and Cyberbullying*). The 2018 edition of the last-mentioned competition [5] brought together 130 teams interested in the topic, 30 teams submitted their test runs and 20 teams also sent their system description paper.

Given that lexical detection methods tend to have low accuracy, other studies tried classical ML approaches [6] or complex classifiers, like SVM, Deep Neural Networks (DNNs) [7], Multi-layer Perceptron (MLP), Bidirectional Encoder Representations from Transformers (BERT), etc. For instance, some studies use crowd-sourced hate speech lexicon to collect texts containing hate speech keywords [8] or explore the effectiveness of Google sentence encoder, Fasttext, Dynamic mode decomposition (DMD) based features and Random kitchen sink (RKS) method [9]. Additionally, previous analysis of hate speech modeling [10] shows that there is a too wide range of features used.

Because lexical aggression can have different tonalities, depending on culture, researchers often subdivided the task into various intercalated categories. One of the most analyzed such language is "hate speech", i.e. discriminative remarks, such as the racist or sexist ones [11]. In this sense, a typology that captures central similarities and differences between subtasks and discuss its implications for data annotation and feature construction has been proposed [12]. The obvious interest in this task of recognizing and distinguishing hate language intensities [13] requires features that capture a deeper understanding of the text not always possible with surface grams.

Summing all existing approaches, we found that the major problem is the continuously changing vocabulary used in offensive texts. Combined with cultural differences, this shaped the first major direction for this paper: to collect a substantial amount of data for Romanian language. Having this data, we used supervised machine learning algorithms to classify comments as offensive or not.

## 3. Dataset

When deciding to develop the offensive language identification system for Romanian, we faced the problem of the corpus. We started thus from and existing annotated corpus of movie reviews, containing 10797 negative comments and 16614 positive reviews. However, negative reviews are not necessarily offensive, the dislike being most of the times expressed in a civilized (although ironical) manner in movie reviews. We updated thus our existing corpus with thousands of YouTube comments, dynamically gathered by a crawling script, and additionally with a few hundred comments extracted from Facebook and Twitch.

A rapid analysis of this quickly obtained corpus found out that the results were largely inconclusive. We therefore started to manually annotate each instance by 3 young annotators, all familiar with the social media style and abbreviations. The main challenge was to have enough context to base our decisions, since many comments are offensive only in a specific context. When the context was either not complete or not clear enough, we removed the comment form the dataset.

In order to evaluate the manually annotation process, we used Cohen's kappa (Cohen's K) [14], a traditional metric that estimates the chance agreement. It represents a correlation coefficient ranged from -1 to +1, where 0 refers to the amount of agreement that can be expected from random chance, while 1 represents the perfect agreement between the annotators. The obtained inter-annotator agreement was of 93%.

Analyzing the collected corpus, it was noticed that some comments from YouTube were pretty long, but only a small section contained offensive language, as in the examples below (shortened due to space restriction):

(RO) (1) huawei nu costa mult, esti *tu prea sarac*. - camera buna - cititorul de amprente destul de rapid (…)
(2) (…) fotografiile sunt mult mai ok fata de p10 sau p10 lite -*prostii* care compara huawei p20 cu iphone x asa compar si skoda cu audi?!
(3) (…) sunetul destul de slab in difuzor, dar nu sunt *cocalar* ca tine sa ascult muzica-n tramvai, deci nu-mi pasa
(EN) (1) huawei doesn't cost much, *you're too poor*. - good camera - fingerprint reader pretty fast –
(2) photos are much better than p10 or p10 lite – *idiots* compare huawei p20 with iphone x why not comparing skoda with audi ?!
(3) the sound is pretty faint in the speaker, but I'm not a *jerk* like you listening to music on the tram, so I don't care.

We considered useful to highlight the offensive part of a longer phrase by using a segmented input approach. Therefore, we first split a comment in text segments and feed them to the algorithm consecutively, thus calculating an offensive score for each segment and a total score. This way, we have a general idea about where in the text the offensive expression is found, and we could provide visual supervised feedback to the user.

Another direction we considered was the Romanian explanatory dictionary. We queried the dictionary to find in the terms' gloss indicators that the word is used in offensive, aggressive, obscene or vulgar context. All the found instanced were manually filtered and the lexicon of bad words and expressions we gathered contains more than 460 items.

Our intuition tells us that non-offensive comments will be classified better than offensive ones due to intentional misspellings or abbreviations/emojis that conceal the meaning, so we want to also make progress in processing this kind of texts. With the corpus growing in offensive words, we were able to identify more spelling mistakes, so this also increased our overall performances.

At the end of this phase, the collected and annotated corpus contains 12,000 offensive and 12,000 non offensive comments, all manually validated.

## 4. Architecture

Figure 1 presents the design flow for the application we developed to test various machine learning algorithms; modules detailed in this section.

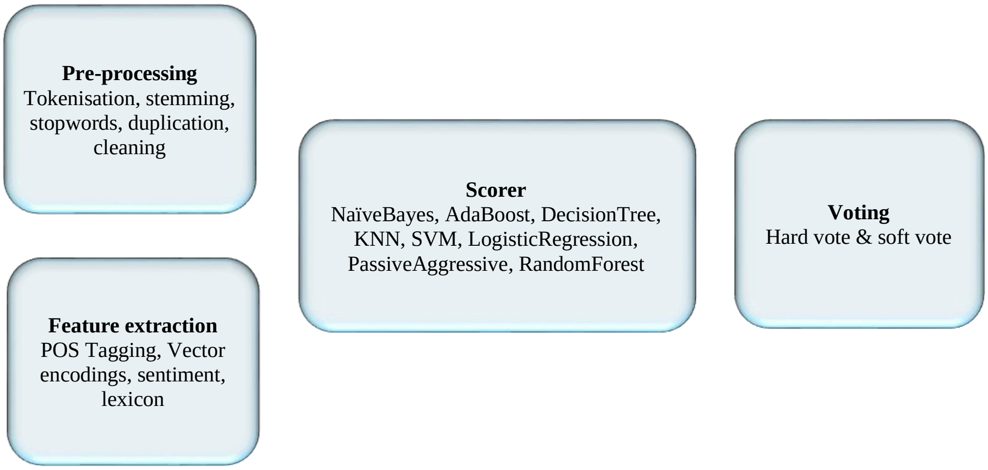
**Pre-processing**
Tokenisation, stemming, stopwords, duplication, cleaning
**Feature extraction**
POS Tagging, Vector encodings, sentiment, lexicon
**Scorer**
NaïveBayes, AdaBoost, DecisionTree, KNN, SVM, LogisticRegression, PassiveAggressive, RandomForest
**Voting**
Hard vote & soft vote

Fig. 1. System architecture.

### *4.1. Pre-processing*

The preprocessing module consists of different methods that transform text and make it computer friendly. The cleaning step removes links, emojis, redundant punctuation marks and stopwords. The text is tokenized and lemmas are identified.

### *4.2. Feature extraction*

An extra layer of POS tagging, using the Romanian TEPROLIN web service was implemented in order to provide additional information. We compared different methods of lemmatization including the snowball lemmatizer and the one TEPROLIN provides us, concluding that the second delivers a better performance.

Another feature considered useful in many articles from the literature was sentiment analysis. We therefore annotated the corpus with polarity (positive or negative sentiments) and used it as a feature for the machine learning algorithms we used.

Our lexicon of more than 460 offensive expressions extracted from the explanatory dictionary of Romanian language was also used as a Boolean feature, indicating if any of the expressions are found or not in the comments.

For encoding the texts into numerical format, we researched language models for Romanian and decided to go with the TF-IDF encoding which has the ability to highlight a word's fitness for a specific context, in direct contrast to the whole dataset.

### *4.3. Scorer*

When choosing the classifiers to be implemented, we closely looked at the ones performing best in the literature. Therefore, AdaBoost and SVM were among the first to implement. Naïve Bayes algorithm, based on applying Bayes' theorem with strong (naive) independence assumptions between the features, was the third option. It is a simple model, but proved to achieve good results in general tasks in a fast and robust manner.

We also tested the tree-like model of decision and consequences, Decision trees, and the clustering technique based on the k-nearest neighbors. Logistic Regression, Passive-Aggressive and Random Forest were also found in the literature and we decided to test them as well for the offensive task on Romanian language.

### *4.4. Voting*

The voting procedure follows the classical hard/soft voting schema. For hard voting, every individual prediction of the considered algorithms is considered, and the class (offensive or not-offensive) with the bigger number of votes establish the label for the test instance. In other words, the predicted label is the one with the highest number of votes among the ones predicted for the instance.

For soft voting, it is not just the label predicted by the algorithms that is used, but also the probability associated with the label. Thus, the prediction for the labels from individual classifiers are weighted by the classifier's importance and the target label with the greatest sum of weighted probabilities wins the vote.

Beside the solution considering all classifiers, we also tested a solution when only the three best performing classifiers were considered. We discovered that soft-best3 works best, but it also is the slowest in terms of running time. The different results we obtained are presented in the next section.

## 5. Results

The developed solution allows us to test the algorithms in the same environment, on the same corpus. Therefore, we initially set the 10-fold validation batches and then run the eight machine learning algorithms 10 times each to make prediction on the test data. As evaluation metrics we choose precision, recall and f-measure, as some of the most widely used evaluation methods.

Four variants were tested for all algorithms: (v1) no pos-tagging, (v2) no lemmatization (v3) only lemmatization, (v4) pos-tagging and lemmatization. As an overall conclusion, it seems that best results were obtained if no pos-tagging was considered, probably due to the very high variability of inflections in Romanian.

Besides each algorithm, we also considered the hard and soft voting for all classifiers, and a hard and soft voting method which only considered the best scoring 3 algorithms.

The first test considered the average running time of each algorithm during both training and test. The results are presented in fig. 2 and 3. As expected, the time for the SVM is longer than the running time of other algorithms, which also influenced the time for the HardVotingAll version.

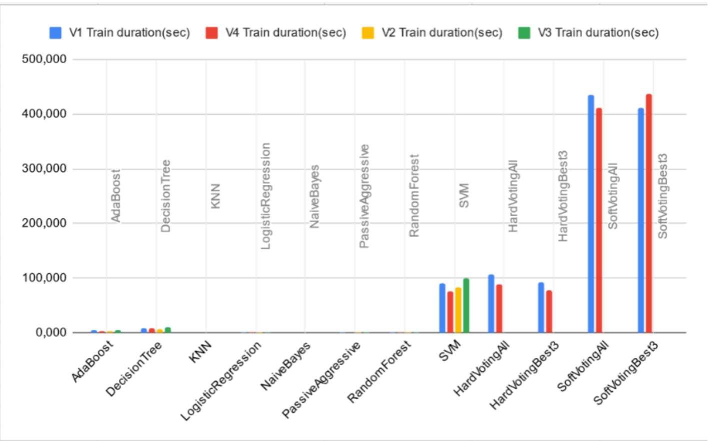

Fig. 2. Running time for training.

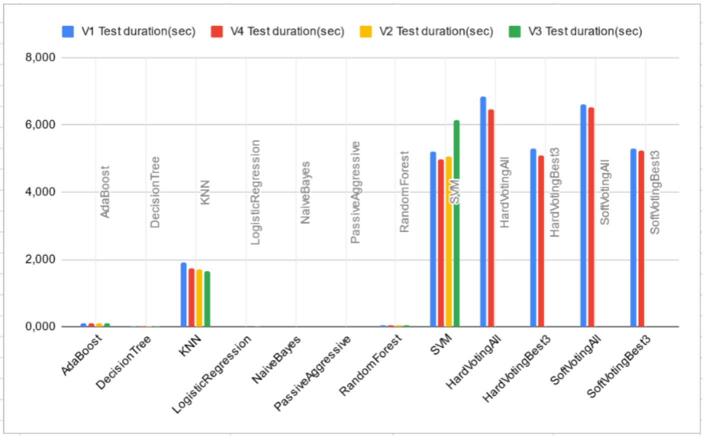

Fig. 3. Running time for testing.

For each of the two labels, we evaluated the eight algorithms and the four voting schemas. We plot the results differentiated on each target label, to better notice the differences.

Thus, when looking at the results obtained by the algorithms for the offensive category (Fig.4), on our corpus of Romanian offensive comments, Random Forest obtained the worst results. Since the variability of the words used in comments greatly varies, and new unseen words are likely to appear at any time, the performance of the bagging-based algorithm is somehow explained. It seems furthermore justified when looking at the second worst performing algorithm, Decision trees, the algorithm Random Forest is based on.

On the other hand, SVMs have proven to be best suited for this task, with performance over 91% for classification of offensive comments.

For the identification of non-offensive comments, Decision Trees, Logistic Regression and KNN proved to obtained the worst results (see figure 5).

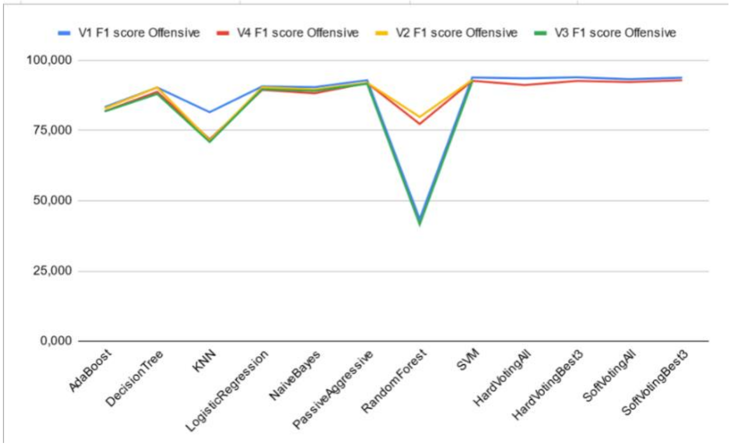

Fig. 4. Performance comparison for the offensive category

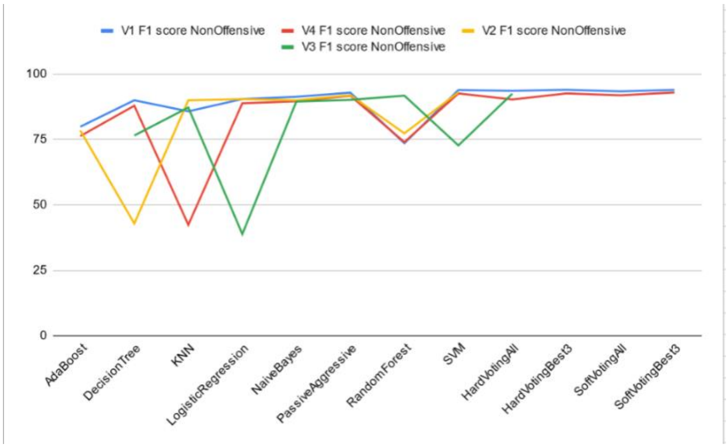

Fig. 5. Performance comparison for the non-offensive category

We calculated the f-measure for each classifier based on multiple values obtained from running the classifiers with different set of features. Table 1 presents the metrics obtained for the best 4 algorithms, in different combinations:

- Having only the TF-IDF score to encode the data.
- Applying TF-IDF on the tokenized data.
- Applying TF-IDF on tokenized and lemmatized data.

While the best precision was obtained when having the data both tokenized and lemmatized, we can notice that the overall best F-measure was obtained only with TF-IDF, although very close to the other ones.

As this study shows, the best overall algorithm for identifying offensive language for Romanian proved to be SVM.

Table 1. Comparative results of various implementations

| | **TF-IDF** | | | **TF-IDF & lemmatising** | | | **TF-IDF & lemmatising & POS Tagging** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Algorithm** | **Precision** | **Recall** | **F1** | **Precision** | **Recall** | **F1** | **Precision** | **Recall** | **F1** |
| Naïve Bayes Offensive | 72.34% | 93.91% | 81.72% | 70.74% | 93.10% | 80.39% | 65.9% | 93.54% | 77.18% |
| Naïve Bayes NonOffensive | 96.58% | 82.75% | 89.13% | 96.32% | 82.41% | 88.82% | 96.81% | 80.07% | 87.65% |
| SVM Offensive | 89.74% | 88.74% | 89.23% | 88.24% | 87.13% | 87.68% | 87.35% | 86.14% | 86.74% |
| SVM NonOffensive | 91.71% | 92.47% | 92.09% | 90.84% | 91.67% | 91.25% | 90.13% | 91.02% | 90.57% |
| PassiveAggressive Offensive | 88.05% | 85.85% | 86.93% | 86.84% | 84.46% | 85.63% | 86.45% | 83.11% | 84.75% |
| PassiveAggressive NonOffensive | 89.43% | 91.13% | 90.27% | 88.78% | 90.57% | 89.66% | 87.66% | 90.21% | 88.91% |
| Logistic Regression Offensive | 88.11% | 87.84% | 87.97% | 86.58% | 86.42% | 86.50% | 84.03% | 86.29% | 85.14% |
| Logistic Regression NonOffensive | 91.12% | 91.32% | 91.22% | 90.44% | 90.56% | 90.50% | 90.62% | 88.98% | 89.79% |

## 6. Conclusions

This study offers information about how various algorithms can enhancing offensive language detection. Furthermore, it is a survey focused on the distinctive nature of online toxicity recognition with the implication of NLP techniques. The results demonstrate that a combination of SVM, Naïve Bayes and Passive-Aggressive can contribute to identify and classify offensive content in social media for the Romanian Language with more then 92%.

As a further work, we are developing an interface to help access social media in a safer manner, by blurring offensive language. We are in the process of integrating a module that detects text zones/ ROI in an image and censors any offensive text.

## Acknowledgements

We thank to the master students form the Computational Linguistics series, as well to 3rd year students from the Faculty of Computer Science, Alexandru Ioan Cuza University of Iaşi, how were involved in the annotation and development of various modules for this system.

## References

[1] Han, Rj., Zeng, Zr., Li, Q. et al. (2020) "Retracted Article: CSI300 Volatility Predicting by Internet Users' Searching Behavior". *Curr Psychol*. https://doi.org/10.1007/s12144-020-00812-2

[2] Patras, Gabriel Florentin, Lungu, Diana Florina, Gîfu, Daniela, Trandabat, Diana (2019) "Hope at SemEval-2019 Task 6: Mining Social Media Language to Discover Offensive Language". *Proceedings of the 13th International Workshop on Semantic Evaluation*, (SemEval-2019), Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019), Minneapolis, USA, pp. 635–638.

[3] Yang, E., & Roberts, M. E. (2021, March). Censorship of online encyclopedias: Implications for NLP models. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 537-548.

[4] El-Alami, Fatima, El Alaoui, Said Ouatik, En Nahnahi, Noureddine (2021) "A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model". *Journal of King Saud University - Computer and Information Sciences*, ISSN 1319-1578. DOI: 10.1016/j.jksuci.2021.07.013 - https://www.sciencedirect.com/science/article/pii/S1319157821001804

[5] Kumar, R., Ojha, A.K., Malmasi, S. and Zampieri, M. (2018) "Benchmarking Aggression Identification in Social Media". In *Proceedings of the 1st Workshop on Trolling, Aggression and Cyberbullying* (TRAC), pp. 1-11.

[6] Nayel, Hamada A., and H. L. Shashirekha (2019) "DEEP at HASOC2019: A Machine Learning Framework for Hate Speech and Offensive Language Detection". In *FIRE* (Working Notes), pp. 336-343.

[7] Sabit, Hassan, Samih, Younes, Mubarak, Hamdy, Abdelali, Ahmed, Rashed, Ammar and Chowdhury, Shammur Absar (2020) "ALT Submission for OSACT Shared Task on Offensive Language Detection." In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 61-65.

[8] Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber (2017) "Automated Hate Speech Detection and the Problem of Offensive Language". In *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1).

[9] Vyshnav, M. T., Sachin Kumar, and K. P. Soman (2020) "Offensive Language Detection: A Comparative Analysis". arXiv preprint arXiv:2001.03131.

[10] Schmidt, A. and Wiegand, M. (2017) "A survey on hate speech detection using natural language processing". In *Proceedings of the 5th International Workshop on Natural Language Processing for Social Media*. Association for Computational Linguistics, Valencia, Spain, pp. 1-10.

[11] Norbata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. (2016) "Abusive Language Detection in Online User Content". In *Proceedings of the 25th International Conference on World Wide Web*, pp. 145–153.

[12] Waseem, Z., Davidson, T., Warmsley, D. and Weber, I. (2017) "Understanding Abuse: A Typology of Abusive Language Detection Subtasks". In *Proceedings of the Abusive Language Online Workshop*, ACL, pp. 78-84. DOI: 10.18653/v1/W17-3012

[13] Malmasi, S., Zampieri, M. (2018) "Challenges in Discriminating Profanity from Hate Speech". *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2): 187-202. Taylor & Francis.

[14] Cohen, J. (1960) "A coefficient of agreement for nominal scales". In *Educational and Psychological Measurement*, 1(20):37–46.

….

**Answers to Review 1:**

We totally agree with the weaknesses noted by the reviewer, which is why we have provided the appropriate explanations, which also highlights the changes made to this paper.

- Indeed, in the paper in the original version we didn’t mention anything about the inter-rater agreement. In fact, we used Cohen's kappa, considered most popular metric to assess the agreement between two annotators (section 3 Dataset).
- We detailed the four variants v1, v2, v3, v4 used in all figures.
- We added some examples of labelled data (section 3 Dataset).
- The text has been completely revised, including formatting, language revision, etc.

**Answers to Review 2:**

Thank you for all your observations so necessary to improve the quality of this paper!

- At this point, we used Cohen's kappa, considered most popular metric to assess the agreement between a pair of annotators (section 3 Dataset).
- The reference 7 is not highlighted in yellow anymore.
- The text has been completely revised, including formatting (e.g., decimals), language revision, typos, references, etc.