The accuracy trap or How to build a phony classifier

Stoll, Anke

Digital
Communication
Research.de

Recommended citation: Stoll, A. (2023). The accuracy trap or How to build a phony classifier. In C. Strippel, S. Paasch-Colberg, M. Emmer, & J. Trebbe (Eds.), *Challenges and perspectives of hate speech research* (pp. 371–381). Digital Communication Research. https://doi.org/10.48541/dcr.v12.22

Abstract: This guide explains, in four steps, how to build a phony text classifier using *supervised machine learning*—a classifier that is absolutely unreliable but looks outwardly sophisticated and attractive. You might enjoy this text if one or more of the following statements apply to you: You are interested in the automated identification of hate speech or related content in online discussions, as long as it looks good; you want to do something with machine learning to impress your peer group, but you do not have the nerve to dig deep into this field as well; you are either a somewhat sneaky or a humorous person. Of course, however, if you are a good and decent researcher, you might also take hints from this text on how *not* to step into the accuracy trap and how not to fall for the tricks of phony classification.

DOI 10.48541/dcr.v12.22

Anke Stoll

The Accuracy Trap or How to Build a Phony Classifier

1 What is this about?

The approach of *Supervised Machine Learning* (SML) raises high expectations for the automated identification of hate speech and all related concepts. These expectations are not surprising, because, after all, *machine learning* is *artificial intelligence*, right? And artificial intelligence is supposed to be groundbreaking. This is already suggested by its very name. And why else would it be so hyped? So, if we do machine learning, build a classifier, train a model, can we not expect groundbreaking results? Of course, as always, it's not that simple. But at least, there is a safe way to build classifiers that *seem* to meet all these expectations—that achieve apparently outstanding results in detecting even the most complex concepts from only text information, based on only a few training instances, and they perform at least as well as the human annotators whose struggle is mirrored in poor values for intercoder agreement.

To build such a phony, hypocritical classification model, we must follow only four simple steps. First, we need a complex and rather elusive concept to identify from text, such as *hate speech*. In the second step, we have to ensure that our sample includes only a few relevant cases. Third, we must stick to the basic estimation functions of SML, including logistic regression or support vector machines.

And, in the fourth step, we must finally use only the standard metric *accuracy* to evaluate the model's performance and avoid any further in-depth performance evaluation if possible.

Following these four steps, we can be quite sure that our classifier will learn nothing about detecting hate speech at all and that it will be absolutely unreliable. But, without a closer look, it will not be noticed since our phony model meets all the expectations at first glance. The following chapters will show you in more detail how to build such a phony classifier that looks nice from the outside but is absolutely useless.

2 Step 1: Choose a contentious concept to classify

Depending on the research area, the SML approach with text data is applied to different issues of interest, including the all-time classics *spam* versus *ham* in emails and *sentiment* in product reviews. The issue of hate speech identification particularly concerns many areas of research and practice and is meanwhile an established research issue in many different fields. In communication studies (and related social sciences), the automated identification of content is not exactly a classical research question (yet), but it is rather relevant from a pragmatic point of view: as an approach of *automated content analysis,* SML is supposed to support the costly and elaborate manual measurement of content for huge amounts of text (e.g., Boumans & Trilling, 2016; Wilkerson & Casas, 2017; Scharkow, 2017; Sommer et al., 2014; Scharkow, 2013). Given its fancy name, the SML method itself is quite a bummer since it is actually just predictive modeling. *Predictive modeling* (and SML, too) means that a dependent variable is supposed to be predicted by a set of independent variables. In SML, the independent variables are called *features*. If the dependent variable is categorical, it is called a *class* or *category*—and the approach is called *classification*. In *text classification* (or *document classification*), the dependent variable is the content or the meaning of a text—for instance, whether a given text includes hate speech. (For a practical introduction to SML, see Müller & Guido, 2017; Géron, 2017).

To identify a text's content or meaning automatically, text characteristics are used as features of prediction. The most essential text characteristics are the words that a text does or does not include. This might sound trivial at first, but it is hard to deny that sentiments, topics, or hate speech are expressed to a significant amount through the use of words. It is therefore reasonable to assume that words can cover some variance of content or meaning. Nevertheless, one can also quickly think of examples where this assumption falls short, where a significant part of the meaning is captured—for example, in the context of the situation or within the person who reads and processes a text. It shows that, for hate speech and its related concepts, this situation often is the case (e.g., Ross et al., 2017; Waseem, 2016).

Overall, a classifier's performance will depend on the modeled statistical relationship between the text features and the text category. When we stick to basic SML (instead of deep learning, for example), that is the statistical relationship between words and meaning, in many cases. But if words have different meanings in different contexts and important variables are missing, that approach starts to weaken. Fortunately, a phony classifier does not have to learn an actual, robust statistical relationship. Ironically, the exact opposite is the case (see also steps 2 and 3 in this guide). For the *hate speech* concept, therefore, one important condition for a phony classifier is fulfilled: an elusive concept supposed to be predicted based on words, even if other important information is probably needed to determine whether a text is considered *hate speech* or not. Luckily, this does not only apply to hate speech but to many concepts that interest communication scholars since they are seemingly impossible to determine without pages of instructions and hours of training (Krippendorf, 2018; Früh, 2015).

3 Step 2: Draw an imbalanced sample with few relevant instances

If we were interested in decent text classification, likely, we would fall over one of the many obstacles that prevent our finding the hoped-for relationship between features (here, words) and meaning (such as hate speech). Many of these obstacles come with the sample itself. But to build a phony classifier, we need not deal with any of these. The only crucial condition is that the sample for training includes just a small portion of relevant cases. Again, hate speech detection is a suitable use case for phony classification since hate speech does not appear in a great number of instances in a random sample of user comments, Tweets, or related text types (e.g., Papacharissi, 2004; Davidson et al., 2017; Coe et al., 2014; Zampieri et al., 2019; Risch et al., 2021; Friess et al., 2021). Since any pattern is, obviously, hard to learn from only a few examples, such *rare event*s are quite unpopular in the whole research field of machine learning (see Haixiang et al., 2017, for an overview). Text data, however, are particularly painful since natural language is a very heterogeneous data type, so text data require huge sample sizes, and it takes forever until patterns form and can be tracked down by any statistical model (Mandelbrot, 1961; Jurafsky & Martin, 2009; Schütze et al., 2008). If we choose a concept such as *hate speech*, however, the heterogeneity problem is somewhat unavoidable since words can express hate in so many ways. Plus, many of the words in hate speech comments are not unambiguously hateful, meaning that they are also used in comments that are not hate speech at all. (See Baden in this volume for an overview of such issues.)

Researchers sometimes address the issues of rare relevant instances and high data heterogeneity by drawing more narrow samples—for example, debates with a certain hashtag, topic, or time span that offer more potential for controversy. Thus, the proportion of relevant cases is often higher, and the text data are not that heterogeneous. In this way, a classifier becomes more likely to learn an actual relationship between words and meaning, though such a classifier would probably not be applicable to other contexts. Luckily, all this struggle is not a problem for phony classification—rather, the opposite applies. We only need one further condition in the sample, which, fortunately, is usually a consequential problem of rare events: an *imbalanced* (unbalanced) derivation of the classes in our sample. *Imbalanced data* here means that comments that include no hate speech are clearly overrepresented (e.g., Stoll, 2020). In conclusion, we are once again blessed with the classification of hate speech since it appears infrequently (at least in manageable sample sizes) and seems rather infrequent compared to instances that do *not* include hate speech.

4 Step 3: Choose a weak classification function

In SML, the classification function models the relationship between features (independent variables) and a category (dependent variable). For text classification, these functions must be able to handle a huge amount of data and a huge number of features at the same time. In SML, a classification function can usually be described as a decision boundary between the instances of Category A (e.g., HATE) and Category B (e.g., NO HATE). Obviously, the classification approach is promising when documents of Category A can be distinguished from documents of Category B, given the words that the text documents (e.g., user comments or Tweets) include. For hate speech, as already discussed in this chapter, this distinction is not always the case since documents of Category A and Category B have many words in common. This problem will most often lead to confused classifiers and unsatisfactory results (Davidson et al., 2017; Waseem & Hovy, 2016).

Luckily, the uncertainty of prediction is a fortune for a phony classifier. In addition to the confusion caused by word overlap between the categories, the small number of relevant instances causes the classifier to finally quit. As Step 2 described, training the classifier on an imbalanced sample where the relevant category (HATE) is underrepresented is a crucial requirement because many classification functions—including *support vector machines*, *logistic regression,* and *decision trees*—tend to predict the major category in the training data in uncertain cases. Indeed, the smaller the data basis, the smaller the chance to find some significant word distribution patterns and the higher the chances that a classifier gives up (e.g., Haixiang et al., 2017; Denil & Trappenberg, 2010; Stoll, 2020). From a statistical perspective, that strategy is straightforward because, in this way, a model will predict the right category in most cases—namely, the overrepresented category in the training data (NO HATE). If we have only 10% of instances annotated as hate, a classifier would be right in 90% of cases if it always predicted NO HATE. However, that rate does not mean hate speech is predicted correctly in 90% of cases. Sometimes, the relevant category HATE will not be predicted at all, which would be an unsatisfactory result, of course, if we were interested in building a model that can actually detect hate speech. During a decent and transparent model evaluation, the scam would be noticed quickly unless we rely only on the popular *accuracy* metric, as the next step describes.

5 Step 4: Stick to the accuracy evaluation metric (only!)

Classification models are usually trained on a subset of an annotated sample, called the *training set*. Then, they are tested and evaluated on a separate sample, called the *test set*. The model performance is measured by how well the predicted values match the true (manually annotated) values on the test set. The most obvious measurement for model quality is the *accuracy*, meaning the *percentage agreement* between the predicted and the annotated values for the dependent variable—here, *hate speech*. The higher the agreement between human annotation and classification, the better. If, for example, only 10% of comments in the sample are hate speech, a classifier could achieve 90% accuracy by only predicting the major category NO HATE without detecting one single hate speech comment. Thanks to a long journey through the method of manual content analysis, communication scholars are already critical of the percentage agreement and prefer the relentless *Krippendorff's alpha*, which also reveals disagreement in rare categories (Krippendorff, 2018; Lombard et al., 2002; Vogelgesang & Scharkow, 2012). Luckily, Krippendorff's paper is yet unknown in the research field of machine learning, so we will probably not be obliged to consider it for reliability measurement. Nevertheless, other established measurements and procedures in the research field of machine learning are quite capable of circumventing the accuracy trap. But do not worry, these other options still do not mean we must give up.

In SML, common measures to evaluate a model are *recall*, *precision*, and the *F1 score*, as a balanced average of both measures (e.g., Powers, 2011). In default setting, all of these measures are used to evaluate a classifier's performance in *one* category, meaning HATE and NO HATE each. A high recall value for the hate category would actually be nice for a hate speech classifier because it would show that many of the instances that have been manually labeled as *hate speech* could have been identified. A phony model, on the contrary, would always have a low recall for the relevant category HATE since it would not really learn how to detect hate speech. Furthermore, acceptable precision in the relevant category would be preferable for an actual hate speech classifier since it would show that the model is not always wrong when it classifies an instance as HATE. A phony model, however, would not learn how to identify hate speech and would, therefore, make many mistakes, which would be reflected in low precision for the category HATE (e.g., Stoll, 2020). So, just reporting recall and precision for the relevant category would be a safe and easy way to expose a phony model. In other words, we certainly do not want to do that! However, if we have followed steps 1 to 3, the recall and precision values for the NO HATE class will most often will be quite nice. To make an impression, these values should be reported in any case, alongside a remarkable accuracy score (how sneaky!).

Not only the evaluation of the test set, which is part of the sample, can reveal an unreliable model. Also, the evaluation on a new data set can be dangerous. Communication studies, meanwhile, have established applying and rechecking a developed instrument for automated content analysis on a completely new data set (e.g., Grimmer & Stewart, 2013). This demand also concerns classifiers—thus, we are still not off the hook. Indeed, this demand is very reasonable since phony models (or models that only learned hate speech from a certain debate) can be exposed without much consideration of in-depth model evaluations. Because we do not want our classifier to be busted, this demand is—of course—annoying. Fortunately, not all data sets considered for external evaluation are actually a cause for concern. If a phony classifier is applied to a new data set, which includes also only a few relevant cases (here, hate*)*, it will be accurate to a high percentage again! Since the model would not have learned to identify hate speech, it would have learned the derivation of hate speech in the training data. If the new data set had a similarly imbalanced derivation, there is nothing to worry about. Good luck!

6 Conclusion

As this guide shows, building a phony classifier that looks outwardly powerful but has learned nothing about hate speech detection at all is fairly simple. Many of the important criteria for phony classification and stepping into the accuracy trap come with the hate speech phenomenon itself. First, people can eternally debate whether a statement should be categorized as *hate speech*—most of all because important information is captured in a context or personal perspective. Second, however, for an ordinary text classifier, none of this information is available, only text. Third, in a random sample of user comments, Tweets, or related data sources, the number of relevant instances from which a machine learning model could learn hate speech is rather small and—in relation to instances that do not include hate speech—rather underrepresented. As a result, classifiers often come out poorly equipped from the training process, having learned hardly more than the imbalanced class derivation in the training data. If we ignore all these flaws, we can still achieve impressive-looking results (see Step 4) that we legitimately expected from something called *machine learning* instead of boring *statistics*. This is because the described circumstances lead to model results, which—

when measured with the right metrics—look like an amazing performance. And at first glance, one could almost think the problem of automated hate speech identification has been effectively solved with logistic regression.

The bad news is that, upon a closer look, the machine learning method *is* just statistics. And, consequently, we are still stuck with the same questions and pitfalls that social scientists already know well enough: *Which information do I need to explain a phenomenon?* versus *These are the independent variables that I am capable of measuring*, or, *Which sample would be suitable for my research questions?* versus *I can only afford a student sample*. Nevertheless, this realization also shows us that machine learning is not far from well-known inferential statistics and, therefore, is predestined to be a further comfort zone for social scientists—only without *p*-values and SPSS.

Anke Stoll is a research associate at the Institute for Social Sciences at the Heinrich Heine University in Düsseldorf, Germany.

References

Boumans, J. W., & Trilling, D. (2016). Taking stock of the toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars. *Digital Journalism*, *4*(1), 8–23. https://doi.org/10.1080/21670811.2015.1096598

Coe, K., Kenski, K., & Rains, S. A. (2014). Online and uncivil? Patterns and determinants of incivility in newspaper website comments. *Journal of Communication*, *64*(4), 658–679. https://doi.org/10.1111/jcom.12104

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, *11*(1), 512-515.

Denil, M., & Trappenberg, T. (2010). Overlap versus imbalance. In A. Farzindar & V. Kešelj (Eds.), *Advances in artificial intelligence. Canadian AI 2010. Lecture notes in computer science* (vol. 6085) (pp. 220–231). Springer. https://doi.org/10.1007/978-3-642-13059-5_22

Friess, D., Ziegele, M., & Heinbach, D. (2021). Collective civic moderation for deliberation? Exploring the links between citizens' organized engagement in comment sections and the deliberative quality of online discussions. *Political Communication, 38*(5), 624–646. https://doi.org/10.1080/10584609.2020.1830322

Früh, W. (2015). *Inhaltsanalyse: Theorie und Praxis* [Content analysis. Theory and practice]. UTB.

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis*, *21*(3), 267–297. https://doi.org/10.1093/pan/mps028

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220–239. https://doi.org/10.1016/j.eswa.2016.12.035

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall.

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage.

Lombard, M., Snyder-Duch, J., & Bracken, C. C. (2002). Content analysis in mass communication: Assessment and reporting of intercoder reliability. *Human Communication Research, 28*(4), 587–604. https://doi.org/10.1111/j.1468-2958.2002.tb00826.x

Mandelbrot, B. (1961). On the theory of word frequencies and on related Markovian models of discourse. *Structure of Language and Its Mathematical Aspects, 12*, 190–219.

Müller, A. C., & Guido, S. (2017). *Einführung in Machine Learning mit Python. Praxiswissen Data Science* [Introduction to machine learning with Python: A guide for data scientists]. O'Reilly Media.

Papacharissi, Z. (2004). Democracy online: Civility, politeness, and the democratic potential of online political discussion groups. *New Media & Society, 6*(2), 259–283. https://doi.org/10.1177/1461444804041444

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63. https://arxiv.org/abs/2010.16061

Risch, J., Stoll, A., Wilms, L., & Wiegand, M. (2021). Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In J. Risch, A. Stoll, L. Wilms, & M. Wiegand (Eds.), *Proceedings of the GermEval 2021 Workshop on the identification of toxic, engaging, and fact-claiming comments. 17th Conference on Natural Language Processing KONVENS 2021* (pp. 1–12). Netlibrary. https://doi.org/10.48415/2021/fhw5-x128

Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., & Wojatzki, M. (2017). Measuring the reliability of hate speech annotations: The case of the European refugee crisis. In M. Beißwenger, M. Wojatzki, & T. Zesch (Eds.), *Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication (Bochum)* (pp. 6–9). Bochumer Linguistische Arbeitsberichte. https://doi.org/10.48550/arXiv.1701.08118

Scharkow M. (2013). *Automatische Inhaltsanalyse* [Automated content analysis]. In W. Möhring & D. Schlütz (Eds.), *Handbuch standardisierte Erhebungsverfahren in der Kommunikationswissenschaft* [Handbook of standardized survey methods in communication science] (pp. 289–306). Springer VS.

Scharkow, M. (2017). Content analysis, automatic. In *The international encyclopedia of communication research methods* (pp. 1–14). John Wiley & Sons. https://doi.org/10.1002/9781118901731.iecrm0043

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval*. Cambridge University Press.

Sommer, K., Wettstein, M., Wirth, W., & Matthes, J. (Eds.). (2014). *Automatisierung in der Inhaltsanalyse* [Automation in content analysis]. Herbert von Halem.

Stoll, A. (2020). Supervised Machine Learning mit Nutzergenerierten Inhalten: Oversampling für nicht balancierte Trainingsdaten [Supervised machine learning with user generated content: Oversampling for imbalanced training data]. *Publizistik*, *65*(2), 233-251.

Vogelgesang, J., & Scharkow, M. (2012). Reliabilitätstests in Inhaltsanalysen: Eine Analyse der Dokumentationspraxis in *Publizistik* und *Medien & Kommunikationswissenschaft* [Reliability tests in content analyses: The documentation of reliability in *Publizistik* and *Medien & Kommunikationswissenschaft*]. *Publizistik*, *57*(3), 333–345. https://doi.org/10.1007/s11616-012-0154-9

Waseem, Z. (2016). Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In *Proceedings of the first workshop on NLP and computational social science* (pp. 38–142). Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-5618

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL student research workshop* (pp. 88-93). https://aclanthology.org/N16-2013.pdf

Wilkerson, J., & Casas, A. (2017). Large-scale computerized text analysis in political science: Opportunities and challenges. *Annual Review of Political Science*, *20*, 529–544. https://doi.org/10.1146/annurev-polisci-052615-025542

Zampieri, M, Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). SemEval-2019 Task 6: Identifying and categorizing offensive language in social media (OffensEval). Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 75–86. https://doi.org/10.18653/v1/S19-2010