Volume 34 — Number 1 — 2020 — ISSN 2190-6858

JLCL
Journal for Language Technology
and Computational Linguistics

Special Issue on Offensive Language

Gastherausgeber Guest edited by
Josef Ruppenhofer, Melanie Siegel, Julia Maria Struß

GSCL Gesellschaft für Sprachtechnologie & Computerlinguistik

Contents

Editorial
Josef Ruppenhofer, Melanie Siegel, Julia Maria Struß i

Reviewer Index vi

COLD: Annotation scheme and evaluation data set for complex offensive language in English
Alexis Palmer, Christine Carr, Melissa Robinson, Jordan Sanders 1

Explaining Offensive Language Detection
Julian Risch, Robin Ruff, Ralf Krestel 29

Automating News Comment Moderation with Limited Resources: Benchmarking in Croatian and Estonian
Ravi Shekhar, Marko Pranjić, Senja Pollak, Andraž Pelicon, Matthew Purver 49

Author Index 80

Impressum

Herausgeber	Gesellschaft für Sprachtechnologie und Computerlinguistik (GSCL)
Aktuelle Ausgabe	Band 34 – 2020 – Heft 1
Gastherausgeber	Josef Ruppenhofer, Melanie Siegel, Julia Maria Struß
Anschrift der Redaktion	Adrien Barbaresi Berlin-Brandenburgische Akademie der Wissenschaften Jägerstr. 22/23 10117 Berlin barbaresi@bbaw.de
ISSN	2190-6858
Erscheinungsweise	2 Hefte im Jahr, Publikation nur elektronisch
Online-Präsenz	www.jlcl.org

Recent years have seen a sharp increase in studies of offensive language (and related notions such as abusive language, hate speech, verbal aggression etc.) as well as of patterns of online behavior such as cyberbullying and trolling. Multiple efforts have been launched for the exploration of computational approaches and the establishment of benchmark datasets for various languages (Basile et al. (2019), Wiegand et al. (2018), Zampieri et al. (2019)).

Although many researchers start out with the intuition that 'I know it when I see it', it turns out that nailing down the boundaries of offensive language and implementing a computational approach for its recognition are abiding challenges. Not surprisingly, the inventory of categories used to classify units of text varies significantly across different papers and shared tasks. The most common division is between OFFENSIVE LANGUAGE (or ABUSE or HATE etc.) and a neutral or other class. Beyond binary classification, there are several ways to deepen the analysis. One of them consists in subdividing the overall offense class according to the targeted group, distinguishing, for instance, sexism and racism from a neither class (Waseem and Hovy (2016)). Another way of adding detail to the analysis is to distinguish sub-classes of offensive language. The 2018 GermEval Shared Task on Offensive Language (Wiegand et al. (2018)), for instance, included a fine-grained task that split up offensive language into the three sub-classes abuse, insult and profanity. A third option is to use a numeric measure of offensiveness (Ross et al. (2016)), foregoing class labels. Yet another dimension along which offensive language can be subdivided is the distinction between explicit and implicit cases (Waseem et al. (2017); Gao et al (2017); Struß el al (2019)). Further dimensions of fine-grained analysis are no doubt conceivable.

Beyond framing the task, sampling a data set is another key problem. As shown by Arango et al. (2019) and Wiegand et al. (2019), many current datasets for abusive language detection contain unwanted biases and artefacts that lead to overly optimistic assessments of the performance of classification systems.

Of the computational approaches taken in designing such systems, there exists a wide and rapidly evolving variety. Schmidt and Wiegand (2017) give a useful recent overview of the state of the art, also providing among other things an extensive discussion of feature-based classification. Wiegand et al. (2018) report that for the GermEval 2018 Shared Task feature-based supervised learning was competitive, even though many neural systems participated and performed well. Struß et al. (2019) report for the GermEval 2019 Shared Task in the following year that supervised classifiers using word embeddings, subword information and ensemble methods, also proved effective. However, similar effectiveness with less task-specific design could be achieved by classifiers based on the BERT model.

Against this background, we had issued a very broad call for contributions to this special issue of JLCL.[1] We are very happy to see that, between them, the contributions

[1] https://easychair.org/cfp/JLCL_SIOL2020

in this issue address a significant range of the topics that we had put forth in our Call for Papers.

The contribution by Palmer et al. presents a new **annotation scheme** for offensive language and hate speech, breaking the difficult annotation task down to four easier to answer questions. Notably, this scheme goes beyond the explicit cases of offensive language and also tackles several types of **complex, implicit, and/or pragmatically-triggered offensive language**. The authors' work also addresses the issue of **evaluation** by providing a new dataset for Evaluation of Complex Offensive Language Data in English (COLD-EN), which in future research can help diagnose systems' ability to appropriately classify instances of a set of special categories of (offensive) language. Among them are reclaimed slurs, non-slur offensive utterances containing pejorative adjectival nominalizations, and utterances conveying offense through linguistic distancing. The authors also conduct some experiments with state-of-the-art classifiers trained on different datasets to evaluate their diagnostic power when it comes to error analysis on offensive language detection.

Under the rubric of **Explainable AI**, the paper by Risch et al. explores how automatic classification systems can be equipped with mechanisms to explain *each individual* classification they make. Transparent and understandable explanations for decisions on which texts to allow or reject are needed to make such systems useful: both content moderators and users need to be able to understand the basis for classifications in order to be able to defend or take issue with them. As speakers' free speech rights hang in the balance and need to be weighed against others' rights to be protected from hateful and discriminatory speech, building explainability into automatic systems would go some way towards alleviating **ethical and legal concerns about automated offensive language detection**.

The third contribution by Shekhar et al. addresses offensive language in a **setting outside of the major Social Media platforms**, exploring the automation of comment moderation for news articles in two **less-resourced languages**, Estonian and Croatian. The authors create new datasets for both languages, labeled in the course process of real world human comment moderation. Owing to the focus on heterogeneous **real-world datasets** and the question of practical applicability, the authors address undesirable content besides offensive language, such as cases of deception and trolling, off-topic posts, copyright infringement, or pornography. The authors provide a systematic comparison of up-to-date classification approaches applied to the data, and propose a number of explanations for differences in performance resulting, for instance, from changes in comments and/or moderation policy over time.

While we could only briefly allude here to some of the contributions contained in these papers, we very much invite you to delve into them further for insight and inspiration. We would also like to thank the colleagues whose work made this special issue possible: the authors of the papers and the reviewers, who contributed to the quality of the published articles with careful and thoughtful feedback: Valerio Basile, Sara Tonelli, Paolo Rosso, Manfred Stede, Alexis Palmer, Torsten Zesch, Tatjana Scheffler, Sylvia Jaki, and Zeerak Waseem. Finally, we want to express our gratitude to the editors of the Journal for Language Technology and Computational Linguistics for their support in putting together this issue, which it is our great pleasure to now release.

The guest editors, Josef Ruppenhofer, Melanie Siegel, Julia Maria Struß.

References

Arango, A., Pérez, J., & Poblete, B. (2019). Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (p. 45–54). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3331184.3331262 doi: 10.1145/3331184.3331262

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., ... Sanguinetti, M. (2019, June). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 54–63). Minneapolis, Minnesota, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/S19-2007 doi: 10.18653/v1/S19-2007

Gao, L., Kuppersmith, A., & Huang, R. (2017). Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach. In *Proceedings of the eighth international joint conference on natural language processing, IJCNLP 2017, taipei, taiwan, november 27 - december 1, 2017 - volume 1: Long papers* (pp. 774–782). Retrieved from https://aclanthology.info/papers/I17-1078/i17-1078

Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., & Wojatzki, M. (2016). 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* (pp. 6–9). Retrieved from https://arxiv.org/pdf/1701.08118.pdf

Schmidt, A., & Wiegand, M. (2017, April). A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media* (pp. 1–10). Valencia, Spain: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W17-1101 doi: 10.18653/v1/W17-1101

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., & Klenner, M. (2019). Overview of GermEval task 2, 2019 shared task on the identification of offensive language. In *Proceedings of the 15th conference on natural language processing (konvens 2019)* (pp. 352–363). Friedrich-Alexander-Universität Erlangen-Nürnberg.

Waseem, Z., Davidson, T., Warmsley, D., & Weber, I. (2017, August). Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the first workshop on abusive language online* (pp. 78–84). Vancouver, BC, Canada: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W17-3012 doi: 10.18653/v1/W17-3012

Waseem, Z., & Hovy, D. (2016, June). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop* (pp. 88–93). San Diego, California: Association for Computational

Linguistics. Retrieved from https://www.aclweb.org/anthology/N16-2013 doi: 10.18653/v1/N16-2013

Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019, June). Detection of Abusive Language: the Problem of Biased Datasets. In *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)* (pp. 602–608). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N19-1060 doi: 10.18653/v1/N19-1060

Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language. In (pp. 1 – 10). Vienna, Austria: Austrian Academy of Sciences. Retrieved from http://nbn-resolving.de/urn:nbn:de:bsz:mh39-84935

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019, June). SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 75–86). Minneapolis, Minnesota, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/S19-2010 doi: 10.18653/v1/S19-2010

Reviewer Index

Valerio Basile
Content Centered Computing
University of Turin
basile@di.unito.it

Sylvia Jaki
Institut für bersetzungswissenschaft und Fachkommunikation
Hildesheim University
jakisy@uni-hildesheim.de

Alexis Palmer
Department of Linguistics
University of North Texas
alexis.palmer@unt.edu

Paolo Rosso
Departamento de Sistemas Informáticos y Computaciòn Universitat Politècnica de València
prosso@dsic.upv.es

Tatjana Scheffler
Department of Linguistics
University of Potsdam
tatjana.scheffler@uni-potsdam.de

Manfred Stede
Department of Linguistics
University of Potsdam
stede@uni-potsdam.de

Sara Tonelli
Digital Humanities research group
Fondazione Bruno Kessler
satonelli@fbk.eu

Zeerak Waseem
Department of Computer Science
University of Sheffield
z.w.butt@sheffield.ac.uk

Torsten Zesch
Language Technology Lab
University Duisburg-Essen, Duisburg, Germany
torsten.zesch@uni-due.de

Alexis Palmer, Christine Carr, Melissa Robinson, Jordan Sanders

COLD: Annotation scheme and evaluation data set for complex offensive language in English

Abstract

This paper presents a new, extensible annotation scheme for offensive language data sets. The annotation scheme expands coverage beyond fairly straightforward cases of offensive language to address several cases of complex, implicit, and/or pragmatically-triggered offensive language. We apply the annotation scheme to create a new **Complex Offensive Language Data Set for English** (COLD-EN). The primary purpose of this data set is to diagnose how well systems for automatic detection of abusive language are able to classify three types of complex offensive language: reclaimed slurs, offensive utterances containing pejorative adjectival nominalizations (and no slur terms), and utterances conveying offense through linguistic distancing.

COLD offers a straightforward framework for error analysis. Our vision is that researchers will use this data set to diagnose the strengths and weaknesses of their offensive language detection systems. In this paper, we diagnose some strengths and weaknesses of a top-performing offensive language detection system by: a) using it to classify COLD, and b) investigating its performance on the 10 fine-grained categories supported by our annotation scheme. We evaluate the system's performance when trained on five different standard data sets for offensive language detection. Systems trained on different data sets have different strengths and weaknesses, with most performing poorly on the phenomena of reclaimed slurs and pejorative nominalizations.
NOTE: *This paper contains sensitive and offensive material. The offensive materials are part of a complex puzzle we wish to better understand; they appear in the form of lightly-censored slurs and degrading insults. We do not condone this type of language, nor does it reflect the attitudes or beliefs of the authors.*

1 Introduction

Existing systems for detecting offensive language[1] mostly rely on identifying explicitly offensive keywords, and they often get things wrong, sometimes in rather troubling ways. Current evaluations tend to misrepresent the capability of these systems to handle the complex nature of offensive language (Wiegand et al., 2019, among others). Systems are particularly challenged when it comes to classification of implicitly offensive

[1] We use "offensive language" as a broad term to include hate speech, abusive language, insults, profanity, and other forms of expression triggering negative reactions in (at least some) hearers/readers.

language. At the same time, systems tend toward false positives on expressions like reclaimed slurs, with outcomes that unfairly and disproportionately penalize certain linguistic and social communities. Mis-classifications of such instances (false positives, in particular) contribute to the systematic racial bias seen in automatic offensive language classification (Davidson et al., 2019). That bias in turn can have a disproportionate negative effect on members of already-stigmatized communities.

In the name of better understanding existing systems' limitations, we present COLD (Complex Offensive Language Dataset),[2] an annotation scheme and evaluation data set that codes for reclaimed slurs as well as two categories of implicitly offensive language: distancing and pejorative adjectival nominalization. We propose the term ***complex offensive language*** to describe occurrences of offensive language signaled with more complex means than explicitly offensive keywords, as well as explicitly offensive keywords used with neutral or positive force. This term encompasses a broad range of linguistic phenomena using different means for conveying offense, or using purportedly offensive language in non-offensive contexts. COLD is easily extensible to include additional types of complex offensive language, and we hope that the community will join us in including data and annotations for additional types of complex offensive language.

Our vision is that this data set can be used to diagnose the strengths and weaknesses of offensive language detection systems, as well as offer a straightforward framework for error analysis. The motivation is similar to targeted evaluation data sets constructed in order to gauge the ability of automated systems to handle particular linguistic concepts. Our approach is also similar in spirit to efforts like the *Build it, Break it, The Language Edition* shared task (Ettinger et al., 2017) or the Winograd Schema Challenges (Kocijan et al., 2020).

Contributions. The contributions of this work are:

1. Linguistically-grounded discussion of several categories of complex offensive language;
2. A straightforward annotation scheme with simple annotation questions leading to fine-grained categorization of instances;
3. A procedure for attempting to mitigate the residual harms of annotation;
4. An evaluation data set with fine-grained categories relevant for detailed error analysis of systems for automatic detection of offensive language;[3] and
5. Diagnostic analysis of how well some existing systems are able to classify these three categories of complex offensive language.

Overview. Our approach to annotation targets three subtypes of offensive language that, we hypothesize, tend to elude correct classification by automatic systems. Each of

[2] In a previous presentation (Carr, Robinson, & Palmer, 2020), we used the name DEIOLD for the same data set.

[3] The data sets, related code, and annotation guidelines are available at: `https://github.com/alexispalmer/cold`.

these subtypes is connected to a significant linguistic literature, discussed in Section 2. The COLD annotation scheme (Section 3) poses four binary questions to annotators and then uses the answers to these questions to associate a single fine-grained category with each instance. We compile a diagnostic corpus from several different data sources (described in Section 4) and annotate it using the new scheme, creating a set of 2016 instances with both detailed linguistic annotation and fine-grained categorization.[4] We train neural models (Section 5) on five different data sets and analyze the models' performance on the new COLD corpus (results in Section 5.3).

2 Selected linguistic phenomena in complex offensive language

Offensive language that uses means other than explicitly offensive slur terms or profanity to convey offense is harder to automatically detect than explicitly offensive language. Such non-explicit phenomena are often referred to as *implicitly offensive language* (Wiegand et al., 2019; Waseem et al., 2017, among others). Another category that poses challenges for automatic detection systems are reclaimed or reappropriated uses of slur terms. Though identical in form with explicit slur terms, they serve entirely different semantic and pragmatic functions (Rahman, 2012). In this paper, we propose the term **complex offensive language** as an umbrella term for the range of linguistic phenomena posing challenges for automated detection of offensive language online.

The space of phenomena constituting the category of complex offensive language has yet to be fully delineated, and we suspect the members of this category are many and varied – offensive language is inherently a complex enterprise. The current work addresses three different phenomena, each of which we find to be both important and linguistically interesting.

The phenomena are realized in COLD as features of the annotation scheme. This section touches on some of the linguistic background for each of these features, each of which is complex and well-studied. We offer only a brief introduction to the literature and encourage the reader to consult cited works for more detailed and nuanced discussion.

First, we consider the phenomenon of **distancing** and several linguistic realizations of distancing (Sec. 2.1). Next, we consider **slurs** in both their typical uses and in the case of slurs reclaimed by the communities they were originally intended to degrade (Sec. 2.2). The third relevant characteristic is pejoration which arises through **adjectival nominalization**, described in Sec. 2.3.

2.1 Pejoration through distancing

Culpeper (1996) describes distancing/othering as a linguistic act through which a speaker creates space between themselves and another person or group. The distinction between **in-group** and **out-group** is crucial for understanding the use of distancing to cause offense. Following Staszak (2009), the in-group is a group the speaker belongs to,

[4] COLD has since been expanded to 2500 instances. Both the smaller and larger versions of the data set can be found in the repository.

and the out-group is a group the listener or some other individual belongs to. Speakers may use language to show that they think their own social group (in-group) is in some way superior to the out-group, or to show that the out-group has negative qualities. The resulting utterances are offensive but not easily detectable by automatic means, particularly when there is no offensive keyword to strengthen pejorative meaning.

Recent computational work has investigated the use of distancing-related features for hate speech classification (Alorainy et al., 2019; Burnap & Williams, 2016), but to our knowledge there is no previous data set which codes for the presence of distancing.

2.1.1 Impoliteness and distancing

The use of linguistic distancing to create offense is, in essence, a form of impoliteness. The study of impoliteness in language examines how the intent of a speaker can cause social disharmony, often by attacking the face of their interlocutor. Following Brown et al. (1987) and others, **face** can be viewed as public self-image, and conversational ideals should preserve face, thereby maintaining social harmony. Positive face represents a speaker's "desire to be liked and appreciated by others."

The use of offensive language creates disharmony in conversation (and in society), and Culpeper (1996) identifies various strategies speakers use to create disharmony; here we focus on the strategy of positive impoliteness. Positive impoliteness is an attack on the hearer's positive face, their desire to be accepted (Bousfield & Locher, 2008). Speakers may use derogatory words (1)[5,6] to target either an individual or a particular characteristic or characteristics seen as negative by the speaker. Speakers may also use taboo words (2) to cause a face attack, or demonstrative pronouns (3) to distance themselves from the targeted group.

(1) Ceasefire? Let's see how long those t*wel h**ds can go without trying to attack Israel then cry to national media when they get popped again ! [HateBase Twitter]

(2) Thinking that i care i don't give a f*ck about what you think ! [HateBase Twitter]

(3) When I talk about those blacks, I really wasn't talking about you. [HateBase Twitter]

Personal pronouns are another potential distancing strategy, as they can be chosen to represent in- and out-groups, emphasizing the polarization and distancing of the targeted group. Alorainy et al. (2019) confirm that pairs of pronouns are more frequent in hate speech than in neutral utterances. In particular, speakers may choose inclusive vs. exclusive pronouns (*we* vs. *they*, *us* vs. *them*, *ours* vs. *theirs*) to create a dichotomy in which the in-group is seen in a positive light and the out-group more negatively (Riggins, 1997). Examples (4) and (5) illustrate:

[5] We have made the choice to censor quoted slurs and profanity throughout the paper by masking initial vowels with asterisks. We also replace usernames with the token @USER.

[6] Throughout, the source of examples is indicated in square brackets. "HateBase Twitter" refers to the corpus presented in Davidson et al. (2017).

(4) Homelessness, is it our problem or someone else's? Granted the homeless are down on their luck and don't really have a choice weather or not they are poor, but that is not my fault. [(Pandey, 2004)]

(5) @USER all them b*tches far as f*ck from us. *We* got el gran p*llo up this way and it serves us well. @USER [HateBase Twitter]

The term **othering** is sometimes used to describe distancing constructions that use metaphor and stereotypes to highlight distance between two social groups, with the targeted group being negatively viewed by the speaker. One classic othering construction has the form 'X is `ADJ` for `NP`', as seen below. Such constructions are implicitly offensive when the `NP` contains a **neutral counterpart**, or non-pejorative correlate (Hom, 2008). This is essentially a non-pejorative alternative for a potential slur term. In (6), *gay* is the neutral counterpart.

(6) He looks straight for a gay man. [AdjNom: Tw.][7]

(7) You're pretty white for a Mexican, no that means you're prettier. [AdjNom: Tw.]

In example (6) the user creates opposition between *straight men* and *gay men* via the properties stereotypically associated with each group. Similarly, in example (7) the speaker creates distance between white people and Mexicans, reflecting stereotypes about skin color and beauty. Othering constructions can also occur with slurs, as in the back-handed compliment seen in example (8). The slur term makes the utterance explicitly offensive and therefore more easily detectable by automated systems.

(8) They are not like other *n*ggers*. [Twitter]

In this case, the speaker suggests that the grammatical subject of the utterance does not embody the negative properties stereotypically associated with the slur term.

2.1.2 Linguistic form, othering, and distancing

Linguistic form, for example the choice of a definite plural instead of a bare plural, also can be used as a means for the speaker to indicate non-membership in a referenced group. Acton (2014) argues that the definite plural indicates non-membership, as in example (9), in which the speaker suggests that they are not a member of the referenced group, *Americans*. This stands in contrast to the bare plural form in example (10), which makes no commitment as to the speaker's membership (or not) in the referenced group.

(9) The Americans love their fast food! [Acton 2014]

(10) Americans love their fast food! [Acton 2014]

Additionally, Acton points out that certain words, such as *gay*, can take on a derogatory meaning when used in the definite plural form. This is because *gay* has associated

[7] Twitter example from the AdjNom corpus (Robinson, 2018), described in Section 4.

social meaning from multiple statements of othering and marginalization. This social meaning adds distancing to non-membership, which can lead to pejorative meaning.

2.2 Slur terms: use and reappropriation

Slurs are lexical items that convey *negative attitudes* or *heavy emotional connotations* towards a social group (Hess, 2019), typically centered around race, nationality, religion, gender, or sexual orientation (Bianchi, 2014). As one of their functions, slurs derograte targeted groups or individuals, and they specifically call up one or more descriptive attributes of the targeted group.

Sometimes slurs are reclaimed by the community they are intended to oppress, through a process known variably as reclamation, (re-)appropriation, and resignification. When reappropriated, slurs shift in meaning, losing their pejorative load (usually, though not always). Reclaimed slurs are used in many different ways by different speakers in the community, often performing complex identity work. Rahman (2012) is an engaging and detailed look at African American communities and reclamation of the n-word. Theoretical approaches to understanding the variable meanings of slur words have developed from content-based analyses (Hornsby, 2001, for example), which struggle to capture the flexibility of slurs with respect to pejorative content, to more pragmatic accounts (Hom, 2008; Bianchi, 2014; McCready & Davis, 2017). The latter accounts have different views on the details of the interaction between contextualized usage and pejorative association, but share the understanding that appropriated in-group uses of slur terms occur in a context which alters the derogatory force of the word. Further, Hornsby, Hom, and Bianchi all convincingly argue that in-group appropriation of slurs pushes back against their derogatory force, by echoing and subverting offensive uses.

Our primary interest in reclaimed slurs for this work is the following: these uses, judged as non-offensive by human moderators, are very frequently incorrectly flagged as offensive or inappropriate by automated content moderation systems. The high frequency of these false positives leaves some speakers in the difficult situation of having to choose between a) being unfairly penalized for utterances that are unproblematic from their perspective (and, importantly, their community's perspective); or b) censoring their preferred use of such constructions for fear of triggering false flags.

Slurs as slurs. In (11), the Twitter user quoted attacks another user, using the term *f*ggot* to degrade the user and to indicate that the speaker sees gay men as inferior to other social groups. Bolinger (2017) proposes that all slurs have five characteristics in common: they are offensive even when the speaker does not intend them to be, they are offensive even to hearers who are not being targeted, they carry derogatory cultural meanings, they do not all appear to be equally offensive on the surface, and they can occur, sometimes, inoffensively (Bolinger, 2017).

(11) @USER fight me 1 on 1 ur choice of game f*ggot see what happens [HateBase Twitter]

(12) @USER not as sick as you m*zzie trash @USER [HateBase Twitter]

Slurs differ from other general pejorative terms (e.g. **sshole*, *sh*thead*). Pejorative lexical items generally target one or more specific individuals on a personal basis (13), and slurs target individuals based on characteristics of the group (Hay, 2013; Blakemore, 2015; Bach, 2018). The slur in (14) is a term that degrades based on characteristics of an entire religion; the degree of offense is much greater than in (13).

(13) No matter what there is always an *sshole wearing a Yankee cap at these games. [HateBase Twitter]

(14) @USER Send all the f*cking m*zzies home, now DOJ come and get me motherf*ckers. [HateBase Twitter]

Reclaimed slurs. In some cases, slurs may undergo a process of reclamation, or appropriation, through which people belonging to the targeted group use the slur in non-derogatory ways within the targeted community (Hess, 2019). As noted above, reclamation serves many different functions for the community, including (among others) expression of a sense of solidarity and companionship, signaling shared identity and socio-cultural experiences, performing the social and political move of "taking back" a word of violence and oppression, subverting social norms, and expressing friendship or other close relationships.

McCready and Davis (2017) offer what they call an invocational account of slurs, whereby the use of a slur term "invoke[s] a preexisting complex of social attitudes and background related to the slurred group." Under this account, the hearer's interpretation of the speaker's intention in using the slur depends crucially on whether the speaker and hearer each belong to the slurred group or the privileged (i.e. non-targeted) group. The speaker/hearer configuration determines whether the slur is used for subordination of the targeted group member, for expression of solidarity, for indicating complicity, or to make an accusation. For example, expression of solidarity may occur when the slur is uttered by one member of the slurred group to another member of the slurred group.

In (15), an African American Twitter user calls his friend *my n*gga* to signify that both belong to the group and, further, that they have a close relationship. The choice of this phrase over other available terms conveying shared identity (such as *brother*) signals that speaker and hearer are especially close (Rahman, 2012, 148).

(15) Imma have to f*ck my n*gga up with some pot brownies [HateBase Twitter]

In appropriation contexts, reclaimed slurs may be used by political organizations or artists to subvert socio-cultural norms (Hom, 2008). The text in example (16) is from a sign held by a woman at a rally for sex workers. Here the speaker subverts sexual norms placed on women by society by using a word that typically conveys negative attitudes about women.

(16) Sl*ts say yes [Text on a sign from an image on Twitter]

In rarer cases, terms that previously were highly-offensive slurs can, through repeated use and resignification, begin to be used in mainstream settings with neutral or even positive meanings. One such example is *queer*. Though some speakers still use this as a slur (17), the reclaiming of the word has progressed far enough that positive in-community uses are not at all unusual, as in (18).

(17) Die f*cking qu**r. [HateBase Twitter]

(18) I'm literally laughing in shock, amazement, joy. In...just everything. In everything! Pop the champagne American qu**rs. We have arrived. [HateBase Twitter]

In some very specific contexts, which Bianchi (2014) describes as "highly-regulated," even speakers outside of the targeted community can use the expression non-offensively. For example, the phrases *Queer Theory* and *Queer Studies* are acceptable in academic settings (Bianchi, 2014), and *queer* is part of the widely-used acronym *LGBTQ(+)*.

2.3 Adjectival Nominalization

The pejorative use of adjectival nominalizations has become salient in public conversations in recent years. As an example, some public figures have come under fire for how they have referenced certain groups. In examples (19) and (20), the speaker uses *gay* and *black* in otherwise neutral contexts, and yet the particular linguistic form of these terms upset people (especially when spoken the way they were spoken).

(19) For the gays out there—ask the gays and ask the people—ask the gays what they think and what they do [Donald Trump]

(20) I have a great relationship with the blacks. I've always had a great relationship with the blacks. [Donald Trump]

Similarly, there was backlash on Twitter during the first GOP debate (August 6, 2015) over the use of the term *illegals*. Below are some examples of reactions to the multiple uses of *illegals* during the debate.

(21) The word "Illegals" will never cease to make me cringe y'all don't even pretend to see them as human beings #GOPDebate [AdjNom:Tw.]

(22) "Illegals" as a noun is SO PROBLEMATIC #GOPDebate [AdjNom:Tw.]

(23) You know when you are using an adjective as a noun you are being racist "felons" "illegals" "Blacks" #GOPDebate [AdjNom:Tw.]

In example (21), the speaker recognizes the dehumanizing effect of the nominalization, even if they do not pinpoint the linguistic source of the problem.[8] The speakers in (22) and (23) both remark that making the adjective *illegal* into a noun and referring to people in that way is offensive and/or racist. The examples above along with many other reactions to such linguistic forms show that people are aware of this subtle linguistic phenomenon, but the reasons behind it may not be widely understood.

Wierzbiecka (1986) explains that when adjectives are nominalized, the new nominal incorporates reference to a generic category or kind based on a prototypical idea of

[8] See Mendelsohn et al. (2020) for computational approaches to analyzing dehumanization in text.

that category. As seen in example (24) below, *blonde*, as an adjective, contains only the semantic adjectival property of hair color. However, as a noun, *a blonde* (25), it has new semantic properties that are associated with the prototypical idea of what a blonde (person) is. These properties include both factual information, such as +HUMAN, and stereotypical information, such as +DUMB and +SEX_OBJECT.

(24) ADJ: Becky has blonde hair. [Constructed]

(25) NOM: Becky is a blonde. [Constructed]

Some adjectives become strongly pejorative when nominalized. Robinson (2018) argues that the adjectives which undergo this transformation often have meanings tied to demographic features of individuals, such as socioeconomics, ethnicity, gender, or sexuality. When these adjectival demographic features are nominalized, the feature is expanded into a generic kind that is based on the prototypical idea of that class of individuals. This process leads to stereotypical properties, very often negative, being tied to the use of these adjectival nominalizations. Consider the contrast below:

(26) ADJ: Becky is a gay woman. [Constructed]

(27) NOM: Becky is a gay. [Constructed]

In example (26), *gay* is simply an adjective describing Becky's sexual orientation. Example (27), on the other hand, conveys negative sentiment toward Becky. Similarly, Dixon et al. (2018) observe the prevalence of identity terms in toxic comments and note that the syntactic frame in which the term appears can influence whether its use is neutral or offensive.

Robinson performs a focused, in-depth study of four adjectival nominalizations: *poor*, *gay*, *female*, and *illegal*. Additionally, she investigates variation between four different forms the nominalizations can take: indefinite singular (*a gay*), definite singular (*the gay*), bare plural (*gays*), and definite plural (*the gays*). Along with the nominalization process, changes to both reference type and linguistic form can influence the amount of pejorative meaning associated with an expression. Note the examples below.

(28) That moment when you realize ALL illegals are technically criminals. [AdjNom:Reddit]

(29) jesus christ i make a joke and now im a gay? [AdjNom:Tw.]

While the generic reference forms (bare plural and definite plural) can still be pejorative such as in example (28), the offensive meaning is enhanced when a specific individual is evoked using an adjectival nominalization, as in example (29). In addition, annotation studies (Palmer et al., 2017) confirm that nominalizations of these four terms are significantly more likely to convey offense than adjectival uses.

We investigate pejorative nominalizations in this work because they are frequent and insidious in their offensiveness. A straight keyword approach is certain to fail, given that these word forms are typically neutral when used adjectivally but offensive when used as nouns.

Summary. Our goal is to construct an evaluation data set that contains roughly equal numbers of offensive and non-offensive instances, with a reasonably balanced distribution across the categories of offensive slur, reclaimed slur, distancing (pejorative or not), and adjectival nominalization (pejorative or not). We select the latter three phenomena because they are difficult to detect automatically. The remainder of the paper describes the resulting data set and its use for targeted evaluation and error analysis.

3 COLD: The annotation scheme

The annotation scheme has two steps. For each instance, a) four Yes-No questions capture different characteristics of the instance; and b) a fine-grained category label is derived based on the answers to the four questions. The scheme can easily be extended to other types of complex offensive language, simply by adding questions and/or fine-grained categories as needed.

The four questions. The four questions to be answered for each instance are:

1. Is it **Offensive**?
2. Is there a **Slur**?
3. Is there an **Adjectival Nominalization**?
4. Is there **Distancing**?

For each question, the only possible answers are Yes and No. This streamlined process does not ask annotators to determine a final category label for a given utterance, only to determine whether the tweet is offensive overall and whether it contains any of the three linguistic phenomena. The approach of breaking a difficult annotation task down into a number of easier questions is inspired by Friedrich and Palmer (2014). Our approach also allows for easy annotation of instances which contain two or more different phenomena, such as a slur plus distancing.

Fine-grained categories. From the answers to the four questions, we deterministically derive a fine-grained category for the utterance (see Table 1). There are 10 possible categories, 5 of them offensive and 5 non-offensive.

In terms of categorization, we allow the presence of a slur in an utterance to take precedence over the other two linguistic features (nominalization and distancing). In other words, if an utterance has been labeled as containing a slur, only two of the ten categories are available. If the utterance is labeled offensive, it gets the category label *OffSlur*. If not, its category is *Reclaimed*. This decision is taken with error analysis in mind, so that utterances with and without slur keywords are in different groups with respect to error analysis. Other researchers using COLD could make a different decision in terms of the grouping of the instances into fine-grained categories.

Offensive utterances without slurs fall into one of four different categories: *OffNom* for offensive nominalizations; *OffDist* for offensive utterances with distancing; *OffBoth* for

Fine-grained category	Off?	Slur?	AdjNom?	Distancing?
Offensive with slur	y	y	y/n	y/n
Offensive with nominalization	y	n	y	n
Offensive with distancing	y	n	n	y
Offensive with both	y	n	y	y
Offensive, other	y	n	n	n
Reclaimed slurs	n	y	y/n	y/n
Nonoffensive with nominalization	n	n	y	n
Nonoffensive with distancing	n	n	n	y
Nonoffensive with both	n	n	y	y
Nonoffensive, no cues	n	n	n	n

Table 1: Fine-grained categories, defined with respect to four (yes-no) annotation questions.

offensive utterances with both nominalization and distancing; and *OffOther* for offensive utterances with none of the three cues. The categories are similar for non-offensive tweets without slurs: *NonNom*, *NonDist*, *NonBoth*, and *NonNone*.

Annotation guidelines and training. Each annotator attended two in-person training sessions with the authors. During the training sessions, important terms and concepts were discussed, including slurs, reclaimed slurs, adjectival nominalization, and distancing. Numerous examples were given to show what would and would not fall into each category. Additionally, the concepts of explicit and implicit offensive language were discussed, with slurs given as an example of explicit offensive language, and adjectival nominalization and distancing as examples of implicit offensive language. Annotators were given access to a set of written annotation guidelines which they could access online at any point in the annotation process.[9] The following are similar to the examples used for training: slurs (30), reclaimed slurs (31), distancing (32), and adjectival nominalizations (33).

(30) stfu! she thinks your ugly and hard to look at! get he f*ck out of chat f*ggot before i block you [HateBase Twitter]

(31) I love all my qu**r boys!! [HateBase Twitter]

(32) Michelle Obama is pretty for a black woman [HateBase Twitter]

(33) The uncultured poors complaining again? Lol [AdjNom: Tw.]

As part of each training session, each annotator completed a test batch with 50 instances, after which the annotators and the authors met to discuss results and disagreements. This is the only time that this sort of calibration and discussion took place. The first version of the annotation scheme used a different approach, asking annotators to select from a set of labels indicating different categories of offensive and non-offensive language. After the initial independently-submitted batches showed low agreement

[9] Our written guidelines are available in the COLD repository.

between annotators and some confusion about the labels, we switched to the 4-question procedure described above. To conclude training, each annotator independently labeled and submitted a final test batch of 25 instances. The elapsed time between training and the start of annotation was roughly one week for most annotators.

4 COLD: The data set

COLD is a collection of instances extracted from pre-existing offensive language data sets. We use the existing labels to guide sampling from the existing corpora, and each instance is then re-annotated using the scheme described above. To compile COLD, we use focused sampling (Wiegand et al., 2019) from pre-existing data sets to collect a balanced number of instances that potentially belong to the categories of offensive with slur, reclaimed (non-offensive) uses of slurs, adjectival nominalizations, and distancing. For the latter two categories in particular, automatic extraction from unrestricted data is difficult, so other strategies are needed. For nominalization, we select data from a hand-collected and annotated corpus. For distancing, we apply heuristic filters based on syntactic patterns. Focused sampling is appropriate precisely because the data set is intended for focused, diagnostic evaluation and error analysis. It is not intended for use as training data.

This section describes the filtering process over existing data sets (Sec. 4.1), the annotation process (Secs. 4.2 and 4.3), and the resulting final corpus (Sec. 4.4).

4.1 Focused filtering

We select 2400 tweets, comments, and posts, balanced across offensive and non-offensive instances, according to the original annotations in the corpora we sample from. The majority of instances (1860) come from the **HateBase Twitter** corpus of Davidson et al. (2017). This is supplemented by 140 instances from the **WaseemHovy** corpus (Waseem & Hovy, 2016), and 400 from Robinson's **AdjNom** corpus (Robinson, 2018).

Slurs and reclaimed slurs. All instances containing slurs or reclaimed slurs come from the HateBase Twitter corpus, which consists of 25,000 tweets annotated as offensive, hate speech, or non-offensive. As the name suggests, all data comes from Twitter and was extracted using the hate speech lexicon from HateBase[10] via the Twitter API. Original annotations were done via crowd sourcing, with each tweet receiving from 4-9 annotations. The resulting distribution is 5 percent hate speech, 76 percent offensive speech, and the rest neither hate speech nor offensive.

We extract instances containing slurs by querying the corpus for a set of 12 slur terms, then filtering for the number of annotators (minimum 3) who labeled the tweet as offensive and selecting the top 500 results. Reclaimed slurs are selected using a set of 10 frequently-reclaimed slur terms, then filtering for the number of annotators who

[10] http://Hatebase.org

labeled the tweet as non-offensive/neither, and taking the top 500 results. In the case of reclaimed slurs, most selected instances have 2-3 offensive labels and 4-6 non-offensive.

Distancing. To select tweets with potential **distancing**, we rely on heuristic patterns over part-of-speech labels; this particular filtering process is the least successful (see Sec. 4.5 for analysis), as these constructions are the most difficult to detect automatically.

We first tag the HateBase Twitter corpus with part-of-speech labels, using the Stanford POS tagger (Toutanova, Klein, Manning, & Singer, 2003). We then search for particular POS patterns, to retrieve othering expressions (of the form '`ADJ PREP NP`') and distancing using personal and/or possessive pronouns (particularly tweets which set up us/them dichotomies). The retrieved expressions are manually inspected, and for othering expressions, we enforce the constraint that the expression includes a non-pejorative correlate (see also section 2.1). From the HateBase corpus, we extract 100 tweets with othering constructions and 260 with distancing using pronouns.

To reach the target of 500 tweets with potential distancing constructions, we sample additional data from the corpus created by Waseem and Hovy (2016). This corpus consists of about 16,000 tweets, annotated as racist, sexist, or neither. We use the same sampling process applied to the HateBase Twitter corpus. As a result, we extract 140 instances of distancing using pronouns.

Adjectival nominalizations. Finally, we include 400 instances of adjectival nominalizations, selected from the data from Robinson (2018). The corpus includes both nominalizations and adjectival uses of four target forms: *poor*, *female*, *gay*, and *illegal*. The nominalizations are distributed across four linguistic forms: indefinite singular, definite singular, bare plural, and definite plural. Each instance is annotated for both pejoration and linguistic form.

The corpus, which was collected for linguistic research on pejorative nominalizations, contains 1742 nominal forms across the 4 target words, plus adjectival uses. Each instance was manually identified and verified, for example, ensuring that occurrences of *female* have human reference. The instances come from a variety of sources, including reddit, Twitter, YouTube videos and commentary, blogs, and news media. Searches were conducted across various platforms on topics likely to produce high levels of the target forms. For example, subreddits such as The_Donald have threads on the topic of immigration. Most of the data was collected during the 2016 US Presidential elections, and many of the topics center around that election and issues central to the election, such as immigration. Many of the blogs, videos, and subreddits searched cater to the Alt Right or to Men's Rights groups such as Men Going Their Own Way (MGTOW). Other data comes from more neutral sources, such as a CNN interview with a business woman sharing her political beliefs about the potential of a female president.

Search engines work well for finding plural forms. Singular forms, however, present more of a challenge, as search results often return adjectival forms rather than singular noun forms. To resolve this issue, common verbs, such as forms of copular *be*, were added after the singular form while searching, resulting in instances like example (34).

(34) Yeah dude being poor happens from time to time, but being A poor is a way of life. LOL [AdjNom:Tw.]

From the adjectival nominalization corpus, we randomly select 300 instances annotated as pejorative and 100 instances annotated as nonpejorative. These are divided evenly across the four target forms, with 75 pejorative and 25 non-pejorative instances for each lexical item.

4.2 Annotation Process

We trained and employed six undergraduate Linguistics majors for this annotation task. With the goal of 2500 annotated instances, each with labels from three different annotators, we assigned each annotator a total of 1250 instances, divided into batches of either 100 or 50 instances each. Annotators were paid $12/hour for their work, and the batches were distributed through online spreadsheets, allowing annotators to work from their preferred locations. Each batch was assigned to three annotators, shuffling the groupings of annotators from batch to batch. For example, batch 1 was assigned to annotators A-B-C, batch 2 to D-E-F, batch 3 to A-B-D, batch 4 to A-C-E, and so forth.

The corpus is compiled from several pre-existing data sets (see Section 4). We distribute instances from the two smaller data sets evenly across the batches.

Mitigation of the residual harms of annotation.[11] Given the highly toxic nature of the data, and the growing evidence suggesting that regular and repeated exposure to such data can be detrimental (Newton, 2019; Simon & Bowman, 2019; Dwoskin, Whalen, & Cabato, 2019), we took measures to mitigate potential negative effects. First, we openly discussed the potential issues with our annotators, encouraging them to work on the data in small chunks of time, to perform self check-ins of their mental states, and to take breaks as necessary. We checked in with them periodically and encouraged them to reach out to us with any concerns. Second, we incorporated hand-selected cute animal videos into the annotation batches, inserting a video link after every 25 instances. The videos were offered as a required step in the annotation process, with link text reminding the annotators (e.g.) “Watch this cute video for a mental health break. It’s required, and you get paid for doing it!” Anecdotally, the annotators have said that they enjoyed watching the videos and that they did provide a welcome break from the difficult data. Our understanding of the effectiveness of this strategy is limited. A careful and rigorous study of the potential harms of annotating toxic data is needed in the future, as well as studying the effectiveness of various strategies for mitigating those harms.

The potential negative impact of extended exposure to toxic data is not to be underestimated, and we feel that some sort of mitigation strategy should be incorporated whenever annotators are asked to take on this kind of task.

[11] Thanks to an anonymous reviewer for the phrase “residual harms of annotation.”

	COLD: 2016 instances	
	Majority Y	**Majority N**
Offensive	952	1064
Contains Slur	1012	1004
Adj. Nom.	500	1516
Distancing	89	1927

Table 2: Distribution of labels in the COLD corpus.

4.3 Inter-annotator agreement and selection of gold-standard labels

We compute agreement between annotators for each of the four annotation questions using the `nltk.metrics.agreement` implementation of Fleiss' *kappa*, as described in (Artstein & Poesio, 2008).

Over the 2016 instances of the COLD corpus, we see good agreement for the categories of Offensiveness (0.61) and Distancing (0.76), and only moderate agreement for the categories of Slur (0.38) and Adjectival Nominalization (0.43). The label distribution for the four categories appears in Table 2. The high agreement seen for Distancing is an artifact of the skewed distribution for this label, as annotators marked very few instances of distancing.

The relatively low agreement we see for Slurs reflects the fact that this is quite a subjective assessment. We chose not to use any sort of lexicon of slurs, leaving annotators to decide for themselves whether an instance contains a slur. Although there is evidence to suggest that demographic factors influence human classification of offensive language (Waseem, 2016), we have not done a detailed study of this interaction. With the exception of race, the demographic profiles of our annotators are somewhat similar. All identify as female, and all are between the ages of 19 and 25. We did not investigate other demographic factors.

Additional training could be helpful for recognizing nominalizations as well, as this distinction relies on linguistic awareness of part-of-speech categories. Overall, inter-annotator agreement for this data set is moderate.

Gold-standard labels. To produce a stable labeled version of the data set, we take a simple majority vote for each annotation question, for each instance. Table 2 summarizes the distribution of labels in the final corpus, across the four annotation questions. Of the 2016 instances in the data set, 47% are labeled offensive, 50% as containing a slur, 25% as containing adjectival nominalizations, and only 4% as containing distancing.

Annotator self-consistency. Four months after the original annotations were finished, we asked three of the original annotators to re-annotate two batches of 50 instances each, in order to measure annotator consistency and reliability for the task. We selected one batch from early in the original annotation process (first 10 batches) and one from later in the process (last 10 batches), selecting batches that were originally labeled by the three annotators in the consistency study. The annotators did not have access to their original data files. Table 3 shows the percentage of instances for which each annotator agreed with their original annotations, four months later. We see high agreement across the board, with the lowest agreement for the question of whether a tweet is offensive.

Annotator	**Offensive?**	**Slur?**	**AdjNom?**	**Distancing?**
Annotator A	87%	96%	92%	92%
Annotator D	87%	96%	96%	92%
Annotator E	94%	96%	99%	97%

Table 3: Consistency check – percentage of instances (out of 100) for which the annotator gave the same label as their original annotation, 4 months later.

4.4 Data set details and clean-up

As released, COLD consists of 2016 triply-annotated instances, meaning we lost nearly 400 of the extracted instances due to various problems with annotations. First, only 2156 of the instances were sent out to the annotators. After collecting, collating, and cleaning up the annotations, we found that the number of annotators per instance varied. 121 instances had labels from only 1 or 2 annotators and were therefore excluded from COLD. 206 instances had labels from 4 or more annotators. For each of these instances, we randomly selected 3 annotators and set the remaining labels aside. Finally, 19 instances were removed due to problems with ID numbers.[12]

After converting the Y/N labels to fine-grained categories (Sec. 3), we see the distribution of categories shown in Table 4. The corpus shows good representation for the categories of offensive with slur, reclaimed slur, offensive with nominalization, non-offensive with nominalization, and non-offensive (no cues). All categories related

[12] The publicly-available version of the data set includes all annotations. A second, slightly-larger version of COLD is also available, with 2500 instances.

Offensive types	#	**Non-offensive types**	#
Offensive with slur	620	Reclaimed slurs	392
Offensive with nominalization	201	Non-off. with nominalization	141
Offensive with distancing	19	Non-off. with distancing	6
Offensive with both	31	Non-off. with both	0
Offensive, no cues	81	Non-off., no cues	525
Total Offensive	952	Total Non-offensive	1064

Table 4: DEIOLD: number of instances per fine-grained category, based on a majority vote of three annotators. (Category definitions in Table 1.)

to distancing are under-represented, in some cases severely. For example, though we extracted 500 instances potentially containing pejorative distancing, annotators only mark 50 of them as offensive.

Clean-up of annotations. Following annotation, the following cleaning steps were taken to prepare the data for analysis. First, labels were modified as necessary for consistency (e.g. removing leading or trailing spaces, changing case). Next, the format of instance IDs was normalized to contain both a data set code and a unique ID number. New line characters in the middle of tweets were replaced with spaces, and duplicate annotations (i.e. same tweet, same annotator) were removed.

4.5 Problems with distancing.

There is a huge discrepancy between the number of instances we extracted as potentially containing distancing constructions ($n = 500$) and the number labeled by the annotators as containing distancing ($n = 89$). This discrepancy could be caused by several different factors: a) a noisy extracting/filtering process; b) annotator misunderstanding and/or confusion; c) poor training of the annotators; or d) faulty conception of the category.

One way of testing some of these factors is to have the same data labeled by the experts who conceived of the category. We perform a small study in which two of the co-authors labeled 98 instances previously extracted as potential distancing constructions. The experts used the four-question annotation process.

Table 5 shows the results. Of the 98 instances, the two experts agree on the distancing label for 56 instances, slightly more than 50%. 18 of the agreed-upon instances are marked by both annotators as *not* containing distancing. The examples below show the four possible label configurations: Yes-Yes, No-No, Yes-No, and No-Yes.

(35) **Agree: Yes** – Hearing these girls in my class talk about cars is some of the lowest IQ discussion I've ever heard. #NotSexist #JustAnObservation [WaseemHovy]

(36) **Agree: No** – Time to see these girls finally cook.... Sink or swim. Sassy won't save you #mkr [WaseemHovy]

(37) **Disagree: Yes, No** – 61% of welfare/government aid is claimed by white people. So y'all black slander is trash now. [HateBase Twitter]

	Offensive?	**Slur?**	**AdjNom?**	**Distancing?**
Agreement	84%	89%	99%	57%
	Distancing →	**AgreeYes**	**AgreeNo**	**Disagree**
Number of Instances		38	18	42

Table 5: Top: Agreement between two experts for 98 instances potentially containing distancing. Bottom: Number of instances per agreement status.

(38) **Disagree: No, Yes** – These girls should know skinny sausages are no fun at all. #mkr [WaseemHovy]

Where the experts disagree, they are remarkably consistent in their disagreements; in 32 of the 42 instances, Expert 1 chooses Yes and Expert 2 chooses No. This finding suggests that each annotator has a coherent notion of what is intended by distancing, but that there are some differences in their respective ideas of the category. Both experts participated in the training process, potentially leading to mixed messages about distancing coming through to the annotators.

Overall, at least one expert chooses the Yes label for about 80% of this randomly-selected subset of potential distancing constructions, but the two expert annotators only agree on distancing for 56/98. This result clearly indicates significant problems with both the extraction process and our understanding of what counts as distancing.

5 Using COLD for diagnostic evaluation

In the remainder of this paper, we put COLD to its intended use: as a diagnostic evaluation data set, to better understand the performance of offensive language detection systems on difficult categories of complex offensive language.

We use one model architecture – a pre-trained BERT (Devlin, Chang, Lee, & Toutanova, 2018) model fine-tuned on an offensive language data set – and train five different versions, each one fine-tuned on a different data set. There is no overlap between the data sets used for fine-tuning the BERT model and those from which COLD is compiled.

This section describes details of the model (Sec. 5.1) and data sets (Sec. 5.2) and results of the diagnostic evaluation (Sec. 5.3).

5.1 Model details

Due to its state-of-the-art performance on many NLP tasks, as well as its success in the Semeval 2019 offensive language detection task (Zampieri et al., 2019b), we use BERT (Devlin et al., 2018) as our base model. The top performing system in the Semeval task fine-tunes a BERT model on the task's training data set (Liu, Li, & Zou, 2019); we take a similar approach. Using this general architecture, we use five different offensive language and hate speech data sets with varying levels of label granularity, and we report performance for each of the five resulting models.

BERT. Bidirectional Encoder Representations for Transformers (BERT) was released by Google Research (Devlin et al., 2018) and has achieved state-of-the-art results on a variety of NLP tasks. The model is based on Vaswani et al. (2017)'s multi-headed transformer model and utilizes a masked language modeling and next sentence training regime. Typical language modeling schemes have the model attempt to predict the next token given a sequence of input tokens and repeat this process over the entire training set, requiring significant training time and compute resources. Masked language modeling, on the other hand, masks a fraction of the input tokens and trains the model on only those that are masked, reducing training time as well as preventing the bidirectional model from leaking information about the token being predicted. In combination with masked language modeling, input sequences are also paired with true and false following sentences so the model is also forced to determine if it is the true next sentence. Finally, all outputs are concatenated with a classification token, so the model can be used for sentence classification (e.g. as offensive or not).

Several pre-trained base BERT models are available. These models for general English can be fine-tuned on task-specific data sets at a substantially-reduced time and compute cost compared to training a new model from scratch. The fine-tuning process performs the masked language modeling step and uses the target label provided by the data set as the concatenated classification token.

Data preprocessing. We follow the preprocessing techniques used by the winning team NULI (Liu et al., 2019) in the SemEval-2019 Task on offensive language detection in social media posts.

1. **Lower case all text** as the model we use is uncased.
2. **Replace urls with "http"**, preventing wordpiece from segmenting URLs into tokens that have likely never been seen in the training data.
3. **Limit consecutive @USER mentions to 3** to reduce redundancy, as users are mapped to a single placeholder token (@USER).
4. **Segment hashtags into component words.** Tweets often utilize hashtags with important keywords strung together, we use the python library `wordsegment` to separate these into their component tokens.
5. **Replace emojis with word descriptions.** Unicode characters for emojis often have poor embedding representations relative to the tokens describing them, so each emoji is converted to its textual description. (😊 → smiley face)

5.2 Fine-tuning data sets.

We perform the same fine-tuning process as Liu et al. (2019) on five different offensive language classification data sets. We use BERT's built-in tokenizer wordpiece, which splits words into their morphological components, and start with the BERT-Large, Uncased (Original) model with 12 transformer blocks, 12 attention heads, and 110 million parameters. We use the default fine-tuning configuration and train for 3 epochs. Of the five data sets, four use binary labels (Off. or Not), and one uses three different labels for offensive utterances. In the case of OLID-2020, the official shared task data is labeled in a semi-supervised fashion with a confidence score, and we follow Fromknecht and Palmer (2020) in choosing a confidence threshold to convert scores into binary labels. The data sets are summarized in Table 6.

Data set	Offensive	Non-Offensive	Total
HASOC_multi	HATE / OFFN / PRFN - 1,143 / 667 / 451	NOT - 3,591	5,852
HASOC_binary	HOF - 2,261	NOT - 3,591	5,852
OLID-2019	OFF - 4,400	NOT - 8,840	13,240
OLID-2020	OFF - 1,426,195	NOT - 5,834,139	7,260,334
Kaggle-Toxic	TOX - 16,225	NOT - 143,346	159,554

Table 6: Label distributions and statistics for fine-tuning data sets. (LABEL - #)

HASOC. The Hate Speech and Offensive Content Identification in Indo-European Languages data set (HASOC) (Mandl et al., 2019) consists of several thousand social media posts from Twitter and Facebook in English, Hindi, and German. Posts are labelled for three separate sub-tasks: (1) binary coarse grained labels hate/offensive (HOF) or NOT, (2) multi-class fine-grained labels (NOT/HATE/OFFN/PRFN), and (3) posts labeled HOF in task 1 are designated as targeted or untargeted hate/offensive language. The labels:

- Hate Speech (HATE) : Text applying negative attributes to an individual due to membership in a group or hateful comments towards groups because of race, politics, sexual orientation, gender, social status, health condition, or similar.
- Offensive Language (OFFN): Posts containing threats of violence or attempts to degrade, dehumanize, or insult an individual.
- Profanity (PRFN) : Language typically deemed inappropriate such as cursing or swearing that is not abusive or insulting in nature.

We fine-tune two BERT models from this data set using English training instances from tasks 1 (HASOC_Binary) and 2 (HASOC_Multi).

OLID-2019 and OLID-2020. The two OLIDs (Offensive Language Identification Dataset) come from two consecutive years of SemEval shared tasks addressing identification and classification of offensive language.

OLID-2019 (Zampieri et al., 2019a) contains thousands of tweets identified using keywords and labelled with a three layer hierarchical annotation scheme for offensive language. The first layer categorizes text as either offensive or not, the second layer specifies whether the offensive language is targeted or not, and the final layer categorizes the target as an individual, group, or other. Language is considered offensive in this data set if it contains insults, threats, or profanity. We use only the first layer annotations to fine-tune a model as a binary classification task.

OLID-2020 contains millions of tweets following the same three layer hierarchical annotation scheme. Unlike OLID-2019, there are no manually-provided gold-standard labels for OLID-2020. Instead, each tweet is labeled with confidence scores (ranging from $0.0 - 1.0$) derived by averaging over the output of a number of supervised systems for detection of offensive language. Task participants are responsible for converting confidence scores to labels. Following Fromknecht and Palmer (2020), we use a threshold of 0.44 for the first annotation layer (offensive or not).

Kaggle-Toxic. The Kaggle-Toxic[13] data set is a collection of Wikipedia Talk page edits shared on Kaggle as a toxic comment classification competition. The texts are posts and replies by Wikipedia contributors discussing page edits that sometimes lead to arguments and toxic behavior. The data set is annotated for six labels that have some overlap, and often a post contains multiple labels. The labels are toxic, severe-toxic, obscene, threat, insult, and identity-hate. The two toxic labels are somewhat ambiguous, and no annotation guidelines were released for this data set. We remove the ambiguous overlap between classes and reduce the problem from a multi-label to a binary classification task of toxic/not. Any text annotated to have any of the original six categories is labeled as TOX, and any text containing none of the original categories is labelled as NOT.

In-domain classification accuracy. Classification accuracy for each model is evaluated on a development set (20% stratified split of the training data used for fine-tuning). On the original labels, development set accuracies are as follows: (a) HASOC_multi: 65.4%; (b) HASOC_binary: 71.2%; (c) OLID-2019: 79.2%; (d) OLID-2020: 97.5% ; and (e) Kaggle-Toxic: 96.2%. Kaggle-Toxic is an outlier due to both the larger amount of data and its skewed distribution, and the high performance for OLID-2020 reflects the much larger amount of training data available. It's worth noting that accuracy on the test data is significantly lower.

Evaluating these models on COLD means not only moving to a new domain for testing, but also testing on mixed-domain data. Thus we can reasonably expect lower overall performance than what we see on an in-domain development set. This cross-domain evaluation setting provides a realistic view of the performance to be expected when applying a system to new data.

5.3 Diagnosis of performance on fine-grained categories

The ten fine-grained categories associated with the instances in COLD allow us to investigate what kinds of labeling decisions each model makes for each type of instance. To use COLD in its intended diagnostic capacity, we compare the distribution of predicted labels for each category with the expected label, given the majority-vote annotations. It is important to note that the set of output labels used by any given model is determined by the labels in the data set used for fine-tuning the model. The OLID-2019 model, for example, outputs the labels OFF and NOT, where the HASOC_binary model outputs HOF and NOT. The output labels of binary models can be mapped straightforwardly to the distinction between offensive and non-offensive

[13] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

		HasocBinary	**OLID-19**	**OLID-20**	**KagTox**
Type		OFF/NOT	OFF/NOT	OFF/NOT	TOX/NOT
OffSlur	620	77% / 23%	85% / 15%	87% / 13%	88% / 12%
OffNom	201	40% / **60%**	**64%** / 36%	**59%** / 41%	40% / **60%**
OffDist	19	68% / 32%	68% / 32%	**79%** / 21%	63%/ 37%
OffBoth	31	45% / 55%	**74%** / 26%	**71%** / 29%	23% / 77%
OffOth	81	74% / 26%	79% / 21%	80% / 20%	77% / 23%
ReclSlur	392	**79%** / 21%	**79%** / 21%	**83%** / 17%	**86%** / 14%
NonNom	141	34% / 66%	43% / 57%	35% / 65%	28% / 72%
NonDist	6	33% / 67%	77% / 23%	67% / 33%	50% / 50%
NonBoth	0	–	–	–	–
NonNone	525	22% / 78%	27% / 73%	28% / 72%	20% / 80%

Table 7: Diagnostic evaluation, training on different data sets and making predictions for COLD. The table shows the percentage of instances in each category assigned each model output label, for 4 binary models.

fine-grained categories. Our current approach to diagnostic evaluation involves looking at the percentage of instances within one COLD category assigned each of the training labels used by the model.

Results for the four binary models appear in Table 7, and results for the one multi-class model appear in Table 8, in both cases with interesting numbers in boldface. For example, the model fine-tuned on OLID-19 predicts OFF for 79% of the 392 instances in COLD’s **ReclSlur** category, providing empirical verification that this model is terrible at accurately labeling reclaimed slurs.

Finding One: All models do a good job of handling the explicit categories, labeling offensive utterances with slurs as offensive in 72-88% of cases. Similarly, non-offensive utterances with no pejoration cues receive non-offensive labels 73-88% of the time, depending on the model. These results are expected; slurs serve as effective keyword features for all models, and the non-offensive cases without pejoration cues rarely lead the model astray.

Finding Two: All models fail at recognizing reclaimed uses of slur terms as non-offensive; these are classified as offensive or toxic 69-86% of the time. For example, (39) is labeled by annotators as N-Y-N-N, yielding the label **reclaimed**, yet all four models classify the tweet as offensive or toxic. Interestingly, the multi-class model labels most of these as Profanity rather than Offensive.

(39) so many pretty b*tches coming to my birthday dinner [HateBase Twitter]

		HasocMulti
Type		HATE/OFF/PROF // NOT
OffSlur	620	05% / 21% / **46%** // 28%
OffNom	201	13% / 10% / 12% // **65%**
OffDist	19	32% / 26% / 10% // 32%
OffBoth	31	16% / 13% / 10% // 61%
OffOth	81	15% / 28% / 15% // **42%**
ReclSlur	392	01% / 04% / **65%** // 31%
NonNom	141	06% / 06% / 07% // 81%
NonDist	6	17% / 00% / 00% // 83%
NonBoth	0	–
NonNone	525	04% / 03% / 05% // 88%

Table 8: Diagnostic evaluation, training on different data sets and making predictions for COLD. The table shows the percentage of instances in each category assigned each model output label, for 1 multi-class model.

Finding Three: Three of four models tend to miss the pejorative meaning associated with adjectival nominalizations , mislabeling them about 60% of the time. OLID-19 and OLID-20, the exceptions, still get about 40% of such cases wrong. The utterance in (40) is clearly anti-gay and is labeled by annotators as Y-N-Y-N, yielding the label **OffNom**. All four models classify the utterance as non-offensive. In other cases, such as (41), non-pejorative nominalizations are considered offensive by most models.

(40) Another huge reason I'm against gays is because their role in politics. [AdjNom]

(41) As a gay teen I can confirm first hand that there are many gays that are depressed. [AdjNom]

The small number of occurrences of distancing identified by annotators prevent us from drawing any strong claims, but we can look at a representative example. (42) is a canonical pejorative othering construction, labeled by annotators as Y-N-N-Y. Only one model (Kaggle-Toxic) classifies it as toxic, and the other three mark it as non-offensive.

(42) You're pretty for a black girl [HateBase Twitter]

5.4 Discussion

Overall, the patterns seen in the analysis are not unexpected, but the structure provided by COLD allows us to easily see which categories are most difficult for current models.

Diagnosis using COLD reveals that, despite strong classification accuracy overall, state-of-the-art models fail hard on implicit offensive language, as well as on reclaimed slurs. The result on reclaimed slurs is particularly informative, as mis-classifications of such instances (false positives, in particular) contribute to the systematic racial bias seen in automatic offensive language classification (Davidson et al., 2019). That bias in turn can have a disproportionate negative effect on members of already-stigmatized communities.

The research community has long been aware of difficulties with detecting implicit and complex offensive language. The new contribution made here is a mechanism for systematic investigation of the strengths and weaknesses of systems for automatically detecting offensive language. Such systematic investigation, coupled with extensive error analysis, holds promise for making dramatic improvements to models for automatic detection of abusive language.

The results are disappointing for the phenomenon of pejoration through distancing, though the source of the problem likely lies in data extraction and filtering, as well as annotation guidelines and training, rather than in the classification process, given that the corpus contains few instances labeled as containing distancing. Even expert annotators found that distancing occurs in fewer than 50% of the instances extracted as potential cases of distancing.

It is our hope that improving performance on these categories of complex offensive language will improve performance for offensive language detection overall. We also hope to identify and add new linguistic phenomena to the data set, perhaps starting our research with the instances in the **Offensive-Other** category.

We find cross-domain classification, as recommended by Wiegand et al. (2019), to be an appropriate setting for performing this analysis, and we hope that domain effects are mitigated at least to some extent by having assembled the corpus from disparate sources. As a next step, we would like to train additional BERT models, this time fine-tuning on HateBase Twitter and the Waseem and Hovy corpus (removing the instances already in COLD). This will show us whether the error patterns change when the model is trained and evaluated within the same domain.

6 Conclusion

This paper presents a new annotation scheme and evaluation data set for offensive language detection in English which incorporates awareness of reclaimed slurs, pejorative adjectival nominalizations, and pejoration through linguistic distancing. In order to achieve better handling of complex offensive language, we argue that these phenomena need to be considered. A system that can make correct predictions for these patterns will be a stronger system overall, as the phenomena included in COLD require more nuanced awareness of the roles of linguistic form, politeness strategies, and sociolinguistic factors. Rather than building a large training corpus, we start by developing an evaluation data set to diagnose whether these patterns are in fact mislabeled by existing systems.

We evaluate the performance of models trained on five different offensive language data sets and find some interesting patterns of correspondence between predicted labels and fine-grained categories. In particular, we show that current models perform poorly on reclaimed slurs and pejorative nominalizations, no matter which data set is used for model fine-tuning.

Looking ahead, we are considering several different directions for extending this work. First, the data has been made publicly available, together with annotation guidelines and model outputs for analysis.[14] We hope to continue expanding the corpus, both with additional data sampled in a less-restrictive way, and with focused data targeting specific linguistic phenomena. We plan to explore the possibility of collaborative data set expansion, deciding on a common format to make it easy for other researchers to contribute to COLD. In addition, we are considering developing a system to generate diagnostic reports for existing systems. We can use the multiple annotations to identify high-agreement (i.e. "easy") and low-agreement (i.e. "difficult") cases, enhancing the quantitative results with examples of errors the system makes across various categories.

Acknowledgments

First and foremost, our thanks to the three anonymous reviewers whose extensive feedback certainly made this article better. Next, thanks to our fantastic annotators: Tiffany Blanchet, Samantha Boyer, Kathryn Denier, Katie Eilerman, Meg Fletcher, and Meghan Renshaw. We have also benefited from helpful discussions with colleagues Patricia Cukor-Avila and Xian Zhang, as well as with students in LING 5550 at the University of North Texas. This work was made possible by a research seed grant from the Dean's Office of the College of Information at UNT.

References

Acton, E. (2014). *Pragmatics and the social meaning of determiners* (Doctoral dissertation, Stanford, CA). Retrieved from `http://www.emich.edu/english/faculty/documents/suthesisacton.pdf`

Alorainy, W., Burnap, P., Liu, H., & Williams, M. L. (2019). "The Enemy Among Us": Detecting Cyber Hate Speech with Threats-Based Othering Language Embeddings. *ACM Trans. Web*, *13*(3).

Artstein, R., & Poesio, M. (2008). Survey Article: Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, *34*(4), 555–596.

Bach, K. (2018). Loaded words: On the semantics and pragmatics of slurs. *Bad Words: Philosophical Perspectives on Slurs*, 60–76.

Bianchi, C. (2014). Slurs and appropriation: An echoic account. *Journal of Pragmatics*, *66*, 35–44.

Blakemore, D. (2015). Slurs and expletives: A case against a general account of expressive meaning. *Language Sciences*, *52*, 22–35.

Bolinger, R. J. (2017). The pragmatics of slurs. *Noûs*, *51*(3), 439–462.

Bousfield, D., & Locher, M. A. (2008). *Impoliteness in language: Studies on its interplay with power in theory and practice* (Vol. 21). Walter de Gruyter.

[14] `https://github.com/alexispalmer/cold`

Brown, P., Levinson, S. C., & Levinson, S. C. (1987). *Politeness: Some universals in language usage* (Vol. 4). Cambridge University Press.

Burnap, P., & Williams, M. L. (2016). Us and them: identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data science*, *5*(1), 11.

Carr, C., Robinson, M., & Palmer, A. (2020). *Improving hate speech detection precision through an impoliteness annotation scheme.* Retrieved from `https://www.linguisticsociety.org/abstract/improving-hate-speech-detection-precision-through-impoliteness-annotation-scheme` (Presentation at Annual Meeting of the Linguistic Society of America)

Culpeper, J. (1996). Towards an anatomy of impoliteness. *Journal of Pragmatics*, *25*(3), 349–367.

Davidson, T., Bhattacharya, D., & Weber, I. (2019). Racial Bias in Hate Speech and Abusive Language Detection Datasets. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 25–35). Florence, Italy: Association for Computational Linguistics.

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

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 67–73).

Dwoskin, E., Whalen, J., & Cabato, R. (2019). *Content moderators at YouTube, Facebook and Twitter see the worst of the web - and suffer silently.* `https://www.washingtonpost.com/technology/2019/07/25/social-media-companies-are-outsourcing-their-dirty-work-philippines-generation-workers-is-paying-price/`. The Washington Post.

Ettinger, A., Rao, S., Daumé III, H., & Bender, E. M. (2017). Towards Linguistically Generalizable NLP Systems: A Workshop and Shared Task. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems* (pp. 1–10). Copenhagen, Denmark: Association for Computational Linguistics.

Friedrich, A., & Palmer, A. (2014). Situation entity annotation. In *Proceedings of LAW VIII-The 8th Linguistic Annotation Workshop* (pp. 149–158).

Fromknecht, J., & Palmer, A. (2020). UNT Linguistics at SemEval-2020 Task 12: Linear SVC with Pre-trained Word Embeddings as Document Vectors and Targeted Linguistic Features. In *Proceedings of SemEval 2020.* (*to appear*)

Hay, R. J. (2013). Hybrid expressivism and the analogy between pejoratives and moral language. *European Journal of Philosophy*, *21*(3), 450–474.

Hess, L. (2019). Slurs: Semantic and pragmatic theories of meaning. *The Cambridge Handbook of the Philosophy of Language.*

Hom, C. (2008). The semantics of racial epithets. *The Journal of Philosophy*, *105*(8), 416–440.

Hornsby, J. (2001). Meaning and uselessness: how to think about derogatory words. *Midwest studies in philosophy*, *25*, 128–141.

Kocijan, V., Lukasiewicz, T., Davis, E., Marcus, G., & Morgenstern, L. (2020). A Review of Winograd Schema Challenge Datasets and Approaches. *arXiv preprint arXiv:2004.13831*.

Liu, P., Li, W., & Zou, L. (2019). NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 87–91). Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., & Patel, A. (2019). Overview of the HASOC Track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation* (p. 14–17). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3368567.3368584` doi: 10.1145/3368567.3368584

McCready, E., & Davis, C. (2017). An invocational theory of slurs. *Proceedings of LENLS*, *14*.

Mendelsohn, J., Tsvetkov, Y., & Jurafsky, D. (2020). A Framework for the Computational Linguistic Analysis of Dehumanization. *Frontiers in Artificial Intelligence*, *2*(15).

Newton, C. (2019). *The Trauma Floor: The secret lives of Facebook moderators in America.* https://www.theverge.com/2019/2/25/18229714/cognizant-facebook-content-moderator-interviews-trauma-working-conditions-arizona. The Verge.

Palmer, A., Robinson, M., & Phillips, K. K. (2017). Illegal is not a noun: Linguistic form for detection of pejorative nominalizations. In *Proceedings of the First Workshop on Abusive Language Online* (pp. 91–100). Vancouver, BC, Canada: Association for Computational Linguistics.

Pandey, A. (2004). Constructing otherness: A linguistic analysis of the politics of representation and exclusion in freshmen writing. *Issues in Applied Linguistics*, *14*(2).

Rahman, J. (2012). The N word: Its history and use in the African American community. *Journal of English Linguistics*, *40*(2), 137–171.

Riggins, S. H. (1997). The rhetoric of othering. *The language and politics of exclusion: Others in discourse*, *8*, 1–30.

Robinson, M. (2018). *A Man Needs a Female like a Fish Needs a Lobotomy: The Role of Adjectival Nominalization in Pejorative Meaning.* Denton, Texas: University of North Texas. Retrieved from `https://digital.library.unt.edu/ark:/67531/metadc1157617/m2/1/high_res_d/ROBINSON-THESIS-2018.pdf` (Master's thesis)

Simon, S., & Bowman, E. (2019). *Propaganda, Hate Speech, Violence: The Working Lives of Facebook's Content Moderators.*

https://www.npr.org/2019/03/02/699663284/the-working-lives-of-facebooks-content-moderators. National Public Radio.

Staszak, J.-F. (2009). Other/otherness. *The International Encyclopedia of Human Geography*.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (pp. 173–180).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).

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. 138–142). Austin, Texas: Association for Computational Linguistics.

Waseem, Z., Davidson, T., Warmsley, D., & Weber, I. (2017). Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Language Online* (pp. 78–84). Vancouver, BC, Canada: Association for Computational Linguistics.

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.* San Diego, California: Association for Computational Linguistics.

Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019). Detection of Abusive Language: the Problem of Biased Datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 602–608). Minneapolis, Minnesota: Association for Computational Linguistics.

Wierzbiecka, A. (1986). What's in a noun? (or: How do nouns differ in meaning from adjectives?). *Studies in Language*, *10*(2), 353–389.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019a). Predicting the type and target of offensive posts in social media. In *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)* (pp. 1415–1420). Minneapolis, Minnesota: Association for Computational Linguistics.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019b). SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 75–86).

Julian Risch, Robin Ruff, Ralf Krestel

Explaining Offensive Language Detection

Abstract

Machine learning approaches have proven to be on or even above human-level accuracy for the task of offensive language detection. In contrast to human experts, however, they often lack the capability of giving explanations for their decisions. This article compares four different approaches to make offensive language detection explainable: an interpretable machine learning model (naive Bayes), a model-agnostic explainability method (LIME), a model-based explainability method (LRP), and a self-explanatory model (LSTM with an attention mechanism). Three different classification methods: SVM, naive Bayes, and LSTM are paired with appropriate explanation methods. To this end, we investigate the trade-off between classification performance and explainability of the respective classifiers. We conclude that, with the appropriate explanation methods, the superior classification performance of more complex models is worth the initial lack of explainability.

1 Explainability and Interpretability

Automatic classification of text happens in many different application scenarios. One area where explanations are particularly important is in the context of online discussion moderation since the users who participate in a discussion usually want to know why a certain post was not published or deleted. On the one hand, comment platforms need to consider automatic methods due to the large volume of comments they process every day. On the other hand, these platforms do not want to lose comment readers and writers by seemingly censoring opinions. If humans moderate online discussions, it is desirable to get an explanation of why they classify a user comment as offensive and decide to remove it from the platform. Thereby, to some extent, moderators can be held accountable for their decisions. They cannot randomly remove comments but need to give reasons — otherwise, users would not comprehend the platform's rules and could not act by them.

Machine learning approaches have proven to be on or even above human-level accuracy for the task of offensive language detection (Wulczyn et al., 2017). A variety of shared tasks fosters further improvements of this classification accuracy, e.g., with focuses on hate speech against immigrants and women (Basile et al., 2019), offensive language (Zampieri et al., 2019; Struß et al., 2019), and aggression (Bhattacharya et al., 2020). As automated text classification applications find their way into our society and their decisions affect our lives (Risch & Krestel, 2018), it also becomes crucial that we can trust those systems in the same way that we can trust other humans.

Machine-learned models, such as models that detect offensive language, should therefore be comprehensible. The field of Explainable AI (XAI) emerged to address this problem by making models interpretable and/or explainable.

Explainable AI is a young and multidisciplinary research area, ranging from machine learning, data visualization, and human-computer interaction to psychology. Researchers distinguish between interpretability and explainability (Lipton, 2018). Interpretability means to convey a mental model of the algorithm to humans. In other words, if a model is interpretable, humans can grasp how its internals work. In contrast, explainability means to explain individual *predictions* of a model, rather than the full model itself. With an explainable model, humans can comprehend the calculation steps that lead from a particular input to a particular output. On the other hand, interpretability enables developers to understand a model's weaknesses and to improve on them.

Some machine learning algorithms, such as decision trees, logistic regression, and naive Bayes, are interpretable by default. However, with an increasing number of features and sophisticated preprocessing, even these simple models lose their interpretability. More complex, non-linear models, such as neural networks and support vector machines (SVMs) with kernels, achieve better accuracy in some tasks but are not interpretable by default. So it might seem that there is a trade-off between accuracy and interpretability.

Explainability is easier to achieve as it is sufficient to explain only single predictions of a model rather than the model itself. There are explainability methods that are specific to machine learning algorithms (model-based) and methods that can be applied to any model (model-agnostic, post-hoc). With many explained predictions of a black-box model, a human's mental model of the algorithm improves. Thereby, explainability can lead to interpretability.

Recently, there is research that is contrary to post-hoc explanation methods. For example, Rudin (2019) states that the focus should be on creating inherently interpretable models rather than retrospectively explaining black-box models. With the General Data Protection Regulation (GDPR)[1] specifying the right to explanations, developing explainable AI systems is inevitable, and we expect the field of Explainable AI to grow in the future. Especially the highly complex neural networks with millions of parameters raise the bar for many natural language processing tasks significantly. At the same time, these models are non-interpretable black boxes. We see a need to make especially these most complex models explainable to ensure trust in them by humans.

In our work, we train a naive Bayes classifier, an SVM, and recurrent neural network models on a dataset of toxic comments. We examine the explanation methods Layer-wise Relevance Propagation (LRP) and Local Interpretable Model-agnostic Explanations (LIME), but also attention layers. Thereby, our study covers a model-based method, a model-agnostic method, and a self-explanatory model. The naive Bayes classifier serves as a baseline. For the evaluation of the explanation methods, we use a word deletion task, the explanatory power index, and t-SNE projections of document vector representations. We discuss the results and find that the explainability methods LRP

[1] `https://eur-lex.europa.eu/eli/reg/2016/679/oj`

and LIME provide explainability beyond the limits of interpretable machine learning algorithms, such as naive Bayes.

Contributions In summary, with the present article, we make the following contributions: First, we provide an overview of explainability methods that can be applied to offensive language detection. Second, we implement a variety of such methods and compare them in different experiments. Finally, we interpret the results, discuss the strengths and weaknesses of the methods, and summarize implications for future work.

Article Outline The remainder of this article is structured as follows: In Section 2, we describe related work on explainability methods and set our work into its context. Section 3 focuses on those methods that we implement for this study and how we train the underlying models for offensive language detection. We evaluate the different methods and discuss the results in Section 4 and 5, before we conclude in Section 6.

2 Related Work

There are two principal ways to achieve explainability: either by using interpretable classifiers or by extending non-interpretable classifiers with explainability methods. The terms explainability and interpretability have no standard definitions in the context of machine learning. When they are not used interchangeably, the distinction is that explainability refers to comprehending individual predictions, whereas interpretability refers to comprehending the decision function (Došilović et al., 2018; Monroe, 2018; Montavon et al., 2017). The terms *local explainability/interpretability* and *global explainability/interpretability* are used to describe this difference (Mohseni et al., 2018; Ribeiro et al., 2016). For the lack of consensus in terminology, we define the terms for this article:

- A decision function f is called explainable, if the decision $f(x)$ for each single input $x \in X$ (in domain X) can be explained in understandable terms to humans.
- A decision function f is called interpretable, if the whole function f (for the whole domain X) can be explained in understandable terms to humans.

For example, in the special case of a text classifier, an attribution-based explanation method might output one score per input feature, e.g., input word. The scores denote how much each input feature contributes to the classifier's decision. Note that interpretability comprises explainability. To this end, interpretability can be derived from explainability by agglomerating explanations. Ribeiro et al. (2016) propose an algorithm to select inputs so that the explanations of the decisions to those inputs give an interpretation of the model. Depending on the domain context of a model, other explanation forms are possible. For example, there are hierarchical explanations, which explain sentiment analysis decisions by considering word interactions (Singh et al., 2019; Tsang et al., 2018; Murdoch et al., 2018).

2.1 Interpretable Classifiers

Simple models are interpretable without any special methods and abstraction because they align with human intuitions. The most popular interpretable models are decision trees since they can easily be visualized and consist of a set of structured decision rules. Explaining a decision tree's prediction is as simple as following the branches that correspond to the input. The most relevant features are closer to the root of the tree. Thereby, the degree of abstraction for the interpretation can be increased simply by pruning the tree.

Another class of interpretable models is based on discrete probabilities. The naive Bayes classifier is interpretable because it uses simple calculations with discrete conditional probabilities. These probabilities can be interpreted as a contribution to the decision made by the classifier. We use this approach as a baseline in our evaluation.

2.2 Sensitivity Analysis and Shapley Values

Sensitivity analysis and Shapley values are two mathematical concepts behind most explainability methods. Sensitivity analysis figures out how sensitive the output $f(x)$ is to a change in the input x. For an infinitesimal change in x, this can be expressed as the gradient ∇f of the decision function f evaluated for the input x. Baehrens et al. (2010) define $-\nabla f(x)$ as the explanation vector. Simonyan et al. (2014) apply sensitivity analysis to explain image classifications made by convolutional neural networks (CNNs) by using the backpropagation algorithm to obtain the gradient. A simple variant of sensitivity analysis that leads to more specific explanations for image classification is gradients multiplied by input (Shrikumar et al., 2017). Explanations by sensitivity analysis cannot be interpreted as: "What input makes the prediction turn out positive?", but rather as: "How to change the input to make the prediction more positive?".

Shapley values have their origin in coalition game theory. They were proposed to assign each player of a coalition game the contribution he or she makes to the overall outcome of the game (Shapley, 1953). The axioms for Shapley values are also desirable properties in the context of explaining a classifier's decision:

1. **Efficiency** The explanation reflects the outcome of the classifier $f(x)$.
2. **Symmetry** Two features that add the same value to the decision $f(x)$ should be equally relevant.
3. **Additivity** If there are multiple decision functions in an ensemble, the final relevance score should match the sum of the scores of the individual functions.
4. **Dummy Player** A feature that does not change the outcome of the classifier should have no relevance.

Shapley values are not used in practice because of their computational costs. Even if feature interactions are neglected, it is infeasible to do the necessary calculations, especially with high-dimensional data, such as word embeddings. Despite not being used often in its pure form, the concept of Shapley values is still relevant. Lundberg and Lee (2017) propose the SHAP framework inspired by Shapley values and show that other explainability methods are approximations of SHAP.

2.3 LRP and LIME

With layer-wise relevance propagation (LRP), Bach et al. (2015) bring the idea of the efficiency axiom of the Shapley values to deep neural networks. However, propagating the output $f(x)$ directly to the input features is complicated for complex decision functions that contain feature interactions and non-linearities, such as those modeled with neural networks. The LRP method makes use of the layered structure of neural networks to break this problem down by distributing the relevance stepwise for each layer in the network. The layer-wise relevance propagation concept defines the constraint that the summed-up relevance scores for each layer are conserved throughout the propagation. This constraint is called *relevance conservation property*.

Ribeiro et al. (2016) propose Local Interpretable Model-agnostic Explanations (LIME). To explain a decision $f(x)$, LIME approximates the local neighborhood of $f(x)$ with an interpretable classifier $g : \{0,1\}^d \to \mathbb{R}$ that serves as an explanation. Remark that g and f do not have the same domain. The domain of g is a binary space with the same dimension as the feature space. Therefore the input to g does only capture the absence or presence of a feature. LIME considers two aspects to choose the best explanation g for $f(x)$. First, g needs to be a good local approximation of f in the local neighborhood of x. Second, the complexity of g should be low to ensure that g is interpretable. To this end, the best explanation for a decision $f(x)$ is the model g that minimizes the unfaithfulness and the complexity of g.

2.4 Other Explainability Methods

Related work on explainability typically discusses image classification. CNNs are very prominent in this domain. Therefore, many CNN-based explainability methods have been developed. One of the first explainability methods for CNNs is DeConvNet (Zeiler & Fergus, 2014). This approach tries to explain decisions by inverting convolution, ReLU operations, and pooling. Applying sensitivity analysis to CNNs by using backpropagation to obtain the gradient leads to similar explanations (Simonyan et al., 2014). Springenberg et al. (2015) describe the differences between DeConvNet and Sensitivity analysis in the aspect of ReLU operations and propose a combination of the approaches called *guided backpropagation*. Kindermans et al. (2018) argue that splitting the input into a signal and a distractor part can lead to clearer explanations. They compare their methods to Sensitivity analysis, DeConvNet, Guided Backpropagation, and LRP.

Similar to LRP is DeepLIFT (Shrikumar et al., 2017). It also backpropagates relevance through neural network layers and complies with the relevance conservation property. Instead of starting with a relevance score that equals the output of the last layer neuron, DeepLift uses the difference to a reference point as an initial relevance score. The explainability method CAM (Class Activation Mapping), proposed by Zhou et al. (2016), uses a special CNN architecture to learn what parts of an image are important for the decision, by considering the outputs of the last convolutional layer. GradCAM extends CAM by combining it with Sensitivity analysis and thereby avoids to retrain the network for explanations, as it is the case with CAM (Selvaraju et al., 2017). The concept of Taylor-type Decomposition was proposed alongside LRP (Bach et al., 2015) and later refined (Montavon et al., 2017). Instead of using relevance messages to propagate the relevance through the layers, first-order Taylor expansions are used to distribute the relevance scores to the next layer. Sundararajan et al. (2016) introduce *integrated gradients*, a way to fulfill the efficiency axiom of Shapley values by integrating over the gradients with respect to modified (counterfactual) inputs.

Murdoch and Szlam (2017) analyze the hidden cell states of an LSTM to construct interpretable rule-based classifiers. This method, called Cell Decomposition, is an explanation method specific to LSTMs. The same authors propose Contextual Decomposition, which does not only explain decisions with relevance scores to single words but also explains phrases and word interactions (Murdoch et al., 2018).

Related work rarely focuses on explanations for text classification. One publication compares human and automatic evaluation of explanation methods for text classification (Nguyen, 2018) and another one describes the application of an attention-based explanation method to a dataset of personal attacks (Carton et al., 2018). Earlier results of our research on explanations for offensive language classification are published in a short paper (Risch, Ruff, & Krestel, 2020).

2.5 Taxonomy of Explainability Methods

We focus on explainability methods that use feature relevance explanations. The first aspect in which explainability methods can differ is whether they use information about the model's structure or not. Layer-wise relevance propagation, for example, is designed to explain decisions made by neural networks and SVMs, as it uses the layered structure and the activation values of hidden layer neurons. Hence LRP is a *model-based* explainability method. Opposed to that, LIME operates on black-box models and does only use the models' input-output pairs to explain decisions. Thus LIME is a *model-agnostic* (or post-hoc) explainability method. Model-agnostic explainability methods often use sampling to approximate the model with another interpretable surrogate model.

Model-based methods can further be distinguished by the approach they are taking to assign relevance scores. Many methods rely on gradients to explain a decision. The simplest of those methods is sensitivity analysis. Guided Backpropagation, integrated gradients, and gradients × input extend this concept. Layer-wise relevance propagation and DeepLIFT have in common that they make use of the efficiency axiom of Shapley values and redistribute a fixed relevance score onto the features. Other methods, like DeConvNet and Cell Decomposition, are very specific to the machine learning algorithms. There are also so-called self-explanatory machine learning algorithms which inherently provide explanations as a side effect of the decision-making process. An example of such a self-explanatory machine learning algorithm is LSTM with an attention mechanism.

3 Explaining Offensive Language Detection

In order to be of practical relevance, automatic offensive language detection tools need to be trusted by users. Trust can only be established if the automatic decisions can be convincingly explained if needed. We implemented several different algorithms for offensive language detection and combined them with different explanation methods. We published our python code for all classifiers, a web application to visualize the explanations, and the training and evaluation procedures on GitHub.[2]

3.1 Classifiers and Explainability Methods

As a baseline, we implement a multinomial naive Bayes text classifier and add explainability based on the inherent conditional probabilities. This classifier is an example of an interpretable machine learning model. Second, we implement an explainable SVM classifier.[3] For multi-class classifications, we use the one-against-all scheme. We generate explanations for SVM decisions with the model-based explainability method LRP and the model-agnostic explainability method LIME. Last but not least, we implement an LSTM with an attention mechanism, which is an example of a self-explanatory model.

3.2 Dataset

There is a variety of datasets annotated for the detection of hate speech (Gao & Huang, 2017), racism/sexism (Waseem & Hovy, 2016) or offensive/aggressive/abusive language (Struß et al., 2019; Kumar et al., 2018). However, most of them are comparably small because of the immense labeling effort. In this article, we use one of the largest annotated datasets in this field, which contains more than 220,000 comments. Google Jigsaw released this dataset as part of a Kaggle challenge on toxic comment classification.[4] It comprises user discussions from talk pages of the English Wikipedia, where each comment can be labeled as *toxic*, *severe toxic*, *insult*, *threat*, *obscene* or *identity hate* (non-exclusive labels). Table 1 shows that the class distribution is strongly imbalanced.

3.3 Training Procedure

The toxic comments dataset represents a multi-label classification problem. Since there are six labels in the dataset, we can think of the naive Bayes and SVM classifiers as six independent binary naive Bayes classifiers, respectively, six independent binary SVMs. The LSTMs have a slightly different architecture for multi-label problems. All labels share the same LSTM layer, but each label has its own independent fully-connected layer

[2] https://hpi.de/naumann/projects/repeatability/text-mining.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[4] https://www.kaggle.com/c/jigsawtoxic-comment-classification-challenge

Table 1: The class distribution of the dataset is strongly imbalanced. The rarest label *threat* is assigned to only 0.3% of the samples.

Class	Frequency
Clean	201,081
Toxic	21,384
Obscene	12,140
Insult	11,304
Identity Hate	2,117
Severe Toxic	1,962
Threat	689

at the last LSTM output. For the Attention LSTM, each label has its own independent attention layer with the following fully-connected layer.

We train the explainable LSTM with TensorFlow[5] and use the *LRP for LSTM*[6] implementation to explain decisions with LRP. We use an Attention LSTM by Yang et al. (2016) with the difference that we only use an attention layer on the word level and no additional sentence level. We choose the regularization term $C = 0.6$ for the SVM. LSTM and Attention LSTM both have a maximum input length of 250, use a 50-dimensional hidden layer for LSTM cells, and are trained with the Adam optimizer for five respectively three epochs. We train custom GloVe word vectors on the corpus of the training set and the unlabeled comments included in the dataset.

4 Evaluation

First, we compare the classification performance of different approaches for offensive language detection. We then describe the experimental setup for the evaluation of their respective explanations. Finally, we discuss the results.

4.1 Classification Performance

Table 2 presents precision, recall, and F1-score of the trained models on the test set. Both LSTM architectures outperform SVM, which in turn outperforms the naive Bayes baseline. We are unable to get good results for the labels *severe_toxic*, *threat*, and *identity_hate* because each of them makes up less than 1% of the dataset. For the evaluation of explanations, we only focus on the *toxic* label as the classifiers perform best on this label.

[5] https://www.tensorflow.org/

[6] https://github.com/ArrasL/LRP_for_LSTM/

Table 2: Precision (P), Recall (R) and F1-score of the classifiers on the toxic comments dataset. Bold font indicates best F1-score per class.

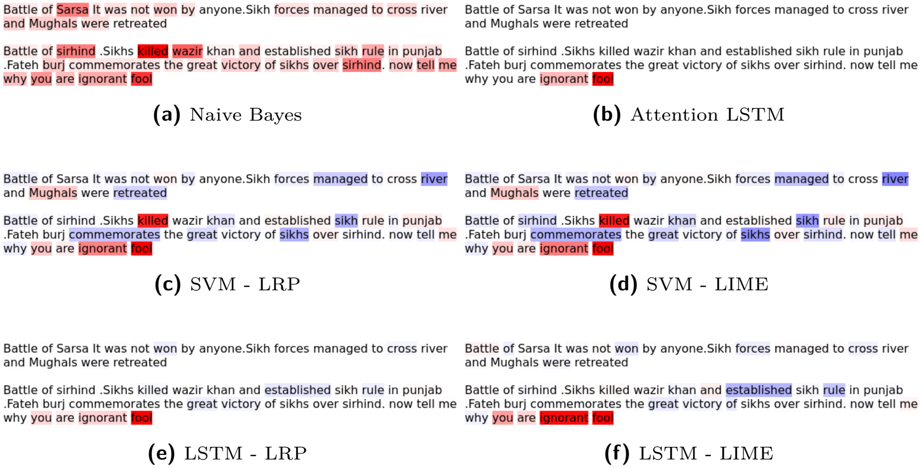
Class Label	Metric	Naive Bayes	SVM	LSTM	Att. LSTM
Toxic	P	69.87	83.22	81.66	84.54
	R	63.89	65.98	68.36	69.74
	F1	66.75	73.60	74.42	**76.43**
Severe Toxic	P	14.45	52.11	56.96	58.33
	R	92.20	18.05	21.95	07.69
	F1	24.98	26.81	**31.69**	13.59
Obscene	P	51.89	85.64	81.09	86.15
	R	75.70	67.57	71.84	67.13
	F1	61.57	75.54	**76.19**	75.46
Threat	P	03.95	72.41	31.43	89.29
	R	59.72	29.17	15.28	35.21
	F1	07.41	41.58	20.56	**50.51**
Insult	P	48.41	78.43	72.67	77.64
	R	75.75	57.82	69.18	59.56
	F1	59.07	66.56	**70.88**	67.40
Identity Hate	P	11.72	64.47	55.36	65.77
	R	73.46	23.22	29.38	49.75
	F1	20.21	34.15	38.39	**56.64**

4.2 Examples of Heatmap Visualization

Explanations by naive Bayes and Attention LSTMs only assign positive relevance scores between 0 and 1. Relevance scores of naive Bayes explanations are probabilities and relevance scores of Attention LSTM explanations are results of a normalizing softmax function. In contrast, explanations by LIME and LRP contain relevance scores that are unbounded and can also be negative. Attention LSTM explanations are the only explanations that are class-independent. Other explainability methods can explain any class, even if the classifier did not predict that class.

The comment in Figure 1 is correctly classified as toxic by both LSTM architectures. The naive Bayes classifier and the SVM classify it as non-toxic.

Figure 1a shows that the naive Bayes classifier explains the toxicity of the comment by marking the word *fool*. The word *killed* is stemmed to *kill* and, therefore, arguably marked also as an explanation, although it is not toxic in this context. Rarely occurring words, such as *Sarsa*, *sirhind*, and *wazir*, are also marked as toxic. The effect of relatively high relevance scores for words that are equally distributed among all classes is amplified in the binary classification case. For a word w that appears with equal



Battle of Sarsa It was not won by anyone.Sikh forces managed to cross river and Mughals were retreated

Battle of sirhind .Sikhs killed wazir khan and established sikh rule in punjab .Fateh burj commemorates the great victory of sikhs over sirhind. now tell me why you are ignorant fool

(a) Naive Bayes

Battle of Sarsa It was not won by anyone.Sikh forces managed to cross river and Mughals were retreated

Battle of sirhind .Sikhs killed wazir khan and established sikh rule in punjab .Fateh burj commemorates the great victory of sikhs over sirhind. now tell me why you are ignorant fool

(b) Attention LSTM

Battle of Sarsa It was not won by anyone.Sikh forces managed to cross river and Mughals were retreated

Battle of sirhind .Sikhs killed wazir khan and established sikh rule in punjab .Fateh burj commemorates the great victory of sikhs over sirhind. now tell me why you are ignorant fool

(c) SVM - LRP

Battle of Sarsa It was not won by anyone.Sikh forces managed to cross river and Mughals were retreated

Battle of sirhind .Sikhs killed wazir khan and established sikh rule in punjab .Fateh burj commemorates the great victory of sikhs over sirhind. now tell me why you are ignorant fool

(d) SVM - LIME

Battle of Sarsa It was not won by anyone.Sikh forces managed to cross river and Mughals were retreated

Battle of sirhind .Sikhs killed wazir khan and established sikh rule in punjab .Fateh burj commemorates the great victory of sikhs over sirhind. now tell me why you are ignorant fool

(e) LSTM - LRP

Battle of Sarsa It was not won by anyone.Sikh forces managed to cross river and Mughals were retreated

Battle of sirhind .Sikhs killed wazir khan and established sikh rule in punjab .Fateh burj commemorates the great victory of sikhs over sirhind. now tell me why you are ignorant fool

(f) LSTM - LIME

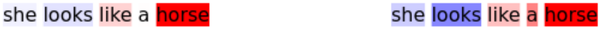
Figure 1: Heatmap visualization of the explanations by the different classifiers and explainability methods. For LRP and LIME, red indicates positive and blue indicates negative relevance.

frequency in both classes c, the relevance of a word is $P(c|w) \approx 0.5$. Together with the unbalanced dataset, this leads to problems for rarely occurring words because the used Laplace smoothing becomes more significant. This smoothing causes high relevance scores for the words *Sarsa*, *sirhind*, and *wazir*. Note that this example comment is labeled as not toxic by the naive Bayes classifier despite the high relevance scores of many words.

Figure 1b shows the explanation generated by the Attention LSTM. The words *fool* and *ignorant* are marked as relevant, and all other words as irrelevant. This explanation aligns with an explanation a human would give. The explanation does not mark *killed* as toxic (in contrast to the naive Bayes classifier). There are two reasons for that. Attention LSTMs do not use stemming (*killed* is considered less toxic than *kill*), and they take into account surrounding words (context awareness).

For toxic comments, we generally observe meaningful explanations by the Attention LSTM. However, for non-toxic comments, Attention LSTMs give misleading explanations. Note that the importance weights that we use as word relevance scores are the result of a softmax function. As a consequence, the Attention LSTM necessarily distributes a relevance score of one among the words — even if there are no toxic elements in the comment. We find that Attention LSTMs often mark punctuation as relevant for non-toxic comments.

Figure 1c and Figure 1d show that LRP and LIME generate almost identical explanations. The toxic words *ignorant* and *fool* are detected by the SVM classifier. The



(a) LSTM - LRP **(b)** LSTM - LIME

Figure 2: Heatmap visualization of the explanations made by LRP and LIME for a contextually toxic comment classified by an LSTM neural network.

word *killed* is also marked as toxic because of stemming. Explanations for non-toxic comments are also very similar for the SVM classifier. The maximum relevance scores for non-toxic comments are much smaller than we would expect.

Explanations for LSTM have an unbalanced relevance distribution among the words. Few words have high absolute relevance and most words have relevance close to zero. These sparse explanations are desirable in the context of this dataset, as there is typically a small set of words that explain the toxicity of a comment.

The LRP explanation is similar to the Attention LSTM explanation, but also includes the word *you*. LIME rates the term *ignorant* as much more toxic than LRP does. We find that LIME often assigns larger negative relevance scores. Explanations for non-toxic comments do not suffer from the problem with the Attention LSTM, as they have much smaller relevance scores overall.

In line with van Aken et al. (2018), we find the labeling of the dataset to be inconsistent. For many comments that are misclassified as toxic, the explanations indeed mark toxic words. Note that the naive Bayes classifier and the SVM label the example comment in Figure 1 as non-toxic, but the explanations highlight the toxic words. The labeling quality of the dataset makes it hard for the classifiers to learn the correct toxicity threshold.

Figure 2 shows a short toxic comment that contains no swear words. The LSTM without attention mechanism is the only classifier that correctly labels this comment as toxic. Without context, none of the words in the comment would be considered toxic on its own. It is therefore difficult to explain this example with attribution-based explanations.

Besides this qualitative evaluation by visualizing the explanations, quantitative, objective methods would be desirable. Unfortunately, evaluating the quality of an explanation is a hard task. Even for human assessors, deciding which explanation is good or bad is very hard and often undecidable. The fact, that the explanations (should) depend on the classification result (which might be wrong), makes the evaluation even more complicated. Nevertheless, we deploy two methods proposed in the literature to automatically and objectively evaluate the generated explanations: word deletion and explanatory power index.

4.3 Word Deletion Task

A good explanation for a text classification characterizes that the words with the highest relevance scores have the most impact on the classification. Therefore deleting relevant words from a text should lead to a significant difference in the classification outcome. The word deletion evaluation measure builds on this premise (Arras et al., 2017). To evaluate a classifier and its explanations with word deletion, we generate explanations for all comments that are correctly classified as toxic (true positives). Consequently, the accuracy on this subset of comments is 100%. We then successively delete words with the highest relevance scores from each text and measure the accuracies of the modified texts at each step. For good explanations, the accuracies should decrease rapidly within the first few word deletions.

And indeed, the accuracy quickly drops in our experiments because only a few words often constitute the toxicity of a comment (e.g., swear words). For all classifiers, more than 80% of the toxic comments could be modified to be not toxic, by deleting only four words. This large number indicates that all classifiers pick up swear words, as those are the explanations for most of the toxic comments. For toxic comments without swear words, the word context is often important, which is the reason for the good performance of LSTMs.

Figure 3 suggests that SVMs give the best explanations according to the word deletion evaluation. This suggestion is misleading because we start for each classifier with its individual subset of true positives: the comments that were correctly labeled as toxic by that classifier. For LSTMs, this subset also contains comments that can only be detected as toxic with word context. It is harder to modify those comments with a few word deletions to be classified as not toxic than it is for comments with a single swear word.

There is no good alternative to using the individual sets of true positives for the evaluation of the explanations for each classifier. In our scenario, the different sets of true positives have a large overlap, which reduces the problem. It is not the case that each classifier is evaluated on entirely different data but rather on slightly different data. We explored the idea of using the intersection of all sets of true positives. However, this approach drastically reduces the size of the dataset for evaluation, and it implicates that the remaining set contains the most simple comments — the ones that *all* classifiers detected correctly as toxic.

4.4 Explanatory Power Index

Arras et al. (2017) propose the Explanatory Power Index (EPI) to evaluate explanations for text categorizations. The method uses an explanation and the corresponding input representation (TFIDF, GloVe word vectors) of a text and combines them to a document summary vector. To obtain this vector, each word in the input representation is scaled by the assigned relevance. Relevant words are emphasized, and irrelevant words are weakened. In the vector space of all input representations, the document summary vectors form clusters of semantically similar texts.

Better explanations lead to better document summary vectors and, therefore, to clearer clusters. The cluster formation can be quantified by the accuracy of a k-nearest neighbor (kNN) classifier that is trained and evaluated on multiple random data-splits

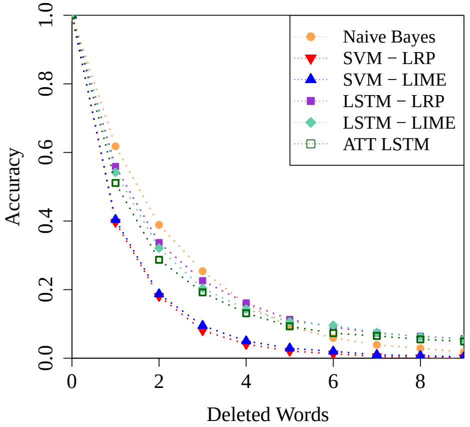


Figure 3: Word deletion experiment for the toxic comments dataset.

of the document summary vectors. The EPI is defined as the mean evaluation accuracy by the kNN classifiers on random data-splits. We use ten splits in our experiments. EPI is decoupled from the predictive power of a classifier since the kNN algorithm is trained and evaluated on the predicted classes for each classifier and not the true classes.

Remark that each entry in a TFIDF vector represents a word. Simply by multiplying each word's vector with the relevance assigned to that word by the explanation, we obtain the document summary vector. In the case of GloVe word vectors, a matrix represents a document, where each row represents a word. Each row vector gets multiplied by the relevance of the corresponding word. In a second step, all row vectors get summed up to obtain the document summary vector. Note that the document summary vector has the same dimension as the word vectors.

EPI uses the accuracy of the kNN classification. To properly use accuracy as a metric, we balance the dataset by downsampling the majority class. We use all toxic comments and randomly sample the same number of non-toxic comments. For each approach, the hyperparameter k is set so that the accuracy (and therefore the EPI) is maximized. Using the baseline representations TFIDF and GloVe word vectors, the kNN algorithm can already distinguish toxic comments from non-toxic comments with high accuracy. The EPI for naive Bayes explanations is worse than for TFIDF. Explanations by naive Bayes often assign high relevance scores to rarely occurring words, which results in the low EPI score. Figure 4c shows that these explanations lead to cluster formations of document summary vectors, but the resulting clusters are not homogeneous.

The explanation methods LIME and LRP have similar EPI scores for the SVM and LSTM classifiers. Figure 4f and Figure 4g confirm these high EPI scores by showing a clear separation of toxic and non-toxic comments into two large clusters. In general, the t-SNE projections of the document summary vectors in Figure 4 suggest that

Table 3: Explanatory Power Index (EPI) for classifiers and explainability methods. Hyperparameter k denotes the number of nearest neighbors that maximizes the EPI.

Classifiers	Explanation	EPI	k
Naive Bayes	Probabilistic	82.29	3
SVM	TFIDF	87.59	25
	LRP	93.38	19
	LIME	93.14	19
LSTM	GloVe	84.74	15
	LRP	**99.67**	3
	LIME	99.48	9
Att. LSTM	Attention	92.04	11

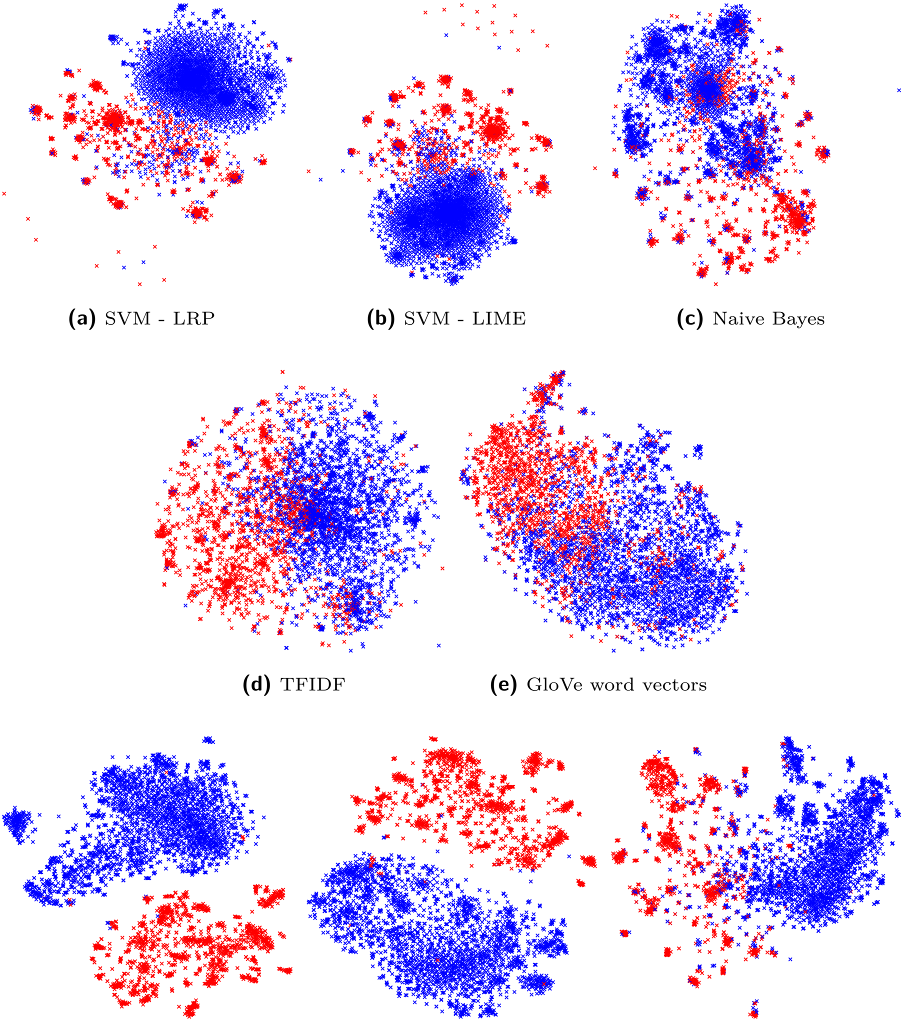
there are multiple clusters of toxic comments. Therefore, document summary vectors could be used to classify and analyze more fine-grained subclasses of toxic comments. The clusters of toxic document summary vectors of the Attention LSTM are denser. However, the separation between the two classes is not as clear as the vectors of the LSTM without an attention mechanism.

5 Discussion

Word deletion and EPI both define quantitative measures to rate explanations, but it is hard to measure the quality of explanations. We defined explainability as the ability to explain a decision of a model in understandable human terms. However, an explanation that aligns with human intuition does not necessarily need to mirror what the model actually is doing. So explainability can only be measured qualitatively within an application context and a target user group. Because our evaluation of explanations is detached from application context and has no target user group, it is hard to rate explanations and explainability methods qualitatively.

The model-agnostic property of LIME comes at the cost of a large number of computations. To achieve stable explanation results with LIME, many perturbed samples need to be classified first. Opposed to that, LRP does a single relevance backpropagation for each explanation. In our experiments, LIME takes up to 40 times longer for explanations than LRP.

The idea to occlude parts of the input and to measure the difference of the output can be generalized beyond text classification and is also used by LIME to generate explanations. Note that LIME is therefore tailored to the word deletion task and might have an unfair advantage in comparison to other explainability methods. For the linear SVM model, LRP and LIME achieve similar results. For more complex decision



(a) SVM - LRP

(b) SVM - LIME

(c) Naive Bayes

(d) TFIDF

(e) GloVe word vectors

(f) LSTM - LRP

(g) LSTM - LIME

(h) Attention LSTM

Figure 4: t-SNE projections of the document summary vectors for each explanation method. For reference, Figures 4d and 4e show t-SNE projections of the TFIDF vectors and GloVe word vectors. Red and blue color mark toxic, respectively, non-toxic comments.

functions, such as those of non-linear LSTMs, the explanations by LRP and LIME differ considerably. All methods by far outperform the interpretable naive Bayes classifier.

The explanations of the self-explanatory LSTM with an attention mechanism have some undesirable characteristics. First, the attention mechanism only explains which words are relevant for a prediction in general (similar to sorting out stop words). However, the relevance scores of the words do not depend on the predicted class. Second, we find that the attention mechanism typically marks only a small set of words as relevant, while all other words are assigned a relevance score close to zero. The attention mechanism was not designed to achieve explainability. We suppose that slight modifications could eliminate the undesirable characteristics. For example, we imagine a hybrid explainability method that uses LRP for the fully-connected layer and the relevance scores of the attention mechanism.

6 Conclusions and Future Work

In this article, we compared four different approaches to make offensive language detection explainable: an interpretable machine learning algorithm (naive Bayes), a model-agnostic explainability method (LIME), a model-based explainability method (LRP), and a self-explanatory model (LSTM with an attention mechanism). We found that LRP and LIME achieve explainability beyond the limits of interpretable algorithms without giving up their superior predictive power.

The model-agnostic method LIME and the model-based method LRP differ mostly in the way they handle negative relevance scores for simple linear models. The attention mechanism of the LSTM cannot provide competitive explanations, which is not surprising, since it was not designed for this task in the first place. However, we assume that the explanatory power of the attention mechanism could be improved by tailoring it to the task of giving explanations.

Last but not least, we find that it is difficult to explain the toxicity of a comment if none of the single words is considered toxic without context. In this case, which includes implicit offensive language, attribution-based explanations fail. Therefore, we see other types of explainability as a promising direction for future work.

References

Arras, L., Horn, F., Montavon, G., Müller, K.-R., & Samek, W. (2017). What is relevant in a text document?: An interpretable machine learning approach. *PLOS ONE*, *12*(8), 1-23.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, *10*(7), 1-46.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research*, *11*, 1803–1831.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F. M. R., … Sanguinetti, M. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@NAACL)* (pp. 54–63).

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., … Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression. *arXiv e-prints*, arXiv:2003.07428.

Carton, S., Mei, Q., & Resnick, P. (2018). Extractive adversarial networks: High-recall explanations for identifying personal attacks in social media posts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 3497–3507).

Došilović, F. K., Brčić, M., & Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 0210–0215).

Gao, L., & Huang, R. (2017). Detecting online hate speech using context aware models. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)* (pp. 260–266).

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., & Dähne, S. (2018). Learning how to explain neural networks: PatternNet and PatternAttribution. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–16).

Kumar, R., Reganti, A. N., Bhatia, A., & Maheshwari, T. (2018). Aggression-annotated Corpus of Hindi-English Code-mixed Data. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC).*

Lipton, Z. C. (2018). The mythos of model interpretability. *Communications of the ACM*, *61*(10), 36–43.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 4768–4777). USA.

Mohseni, S., Zarei, N., & Ragan, E. D. (2018). A Survey of Evaluation Methods and Measures for Interpretable Machine Learning. *arXiv e-prints*, arXiv:1811.11839.

Monroe, D. (2018). AI, explain yourself. *Communications of the ACM*, *61*(11), 11–13.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., & Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, *65*, 211–222.

Murdoch, W. J., Liu, P. J., & Yu, B. (2018). Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–15).

Murdoch, W. J., & Szlam, A. (2017). Automatic Rule Extraction from Long Short Term Memory Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–12).

Nguyen, D. (2018). Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the Conference of the North American Chapter* *of the Association for Computational Linguistics (NAACL)* (pp. 1069–1078).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 1135–1144).

Risch, J., & Krestel, R. (2018). Delete or not delete? semi-automatic comment moderation for the newsroom. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@COLING)* (p. 166-176).

Risch, J., Ruff, R., & Krestel, R. (2020). Offensive language detection explained. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@LREC)* (p. 137-143).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*, 206–215.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the International Conference on Computer Vision (ICCV)* (p. 618-626).

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, *2*(28), 307–317.

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 3145–3153).

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–8).

Singh, C., Murdoch, W. J., & Yu, B. (2019). Hierarchical interpretations for neural network predictions. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–11).

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. A. (2015). Striving for simplicity: The all convolutional net. In *Workshop Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–14).

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., & Klenner, M. (2019). Overview of germeval task 2, 2019 shared task on the identification of offensive language. In *Proceedings of the Conference on Natural Language Processing (KONVENS)* (pp. 354–365).

Sundararajan, M., Taly, A., & Yan, Q. (2016). Gradients of Counterfactuals. *arXiv e-prints*, arXiv:1611.02639.

Tsang, M., Sun, Y., Ren, D., & Liu, Y. (2018). Can I trust you more? Model-Agnostic Hierarchical Explanations. *arXiv e-prints*, arXiv:1812.04801.

van Aken, B., Risch, J., Krestel, R., & Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the Workshop on Abusive Language Online (ALW)* (pp. 33–42).

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).

Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the International Conference on World Wide Web (WWW)* (pp. 1391–1399).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (pp. 1480–1489).

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). In *Proceedings of the International Workshop on Semantic Evaluation (SemEval@NAACL)* (pp. 75–86).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)* (pp. 818–833).

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2921–2929).

Ravi Shekhar, Marko Pranjić, Senja Pollak, Andraž Pelicon, Matthew Purver

Automating News Comment Moderation with Limited Resources: Benchmarking in Croatian and Estonian

Abstract

This article describes initial work into the automatic classification of user-generated content in news media to support human moderators. We work with real-world data — comments posted by readers under online news articles — in two less-resourced European languages, Croatian and Estonian. We describe our dataset, and experiments into automatic classification using a range of models. Performance obtained is reasonable but not as good as might be expected given similar work in offensive language classification in other languages; we then investigate possible reasons in terms of the variability and reliability of the data and its annotation.

1. Introduction

This article describes initial work on the EMBEDDIA project[1] into the automatic classification of user-generated content (UGC) in news media: reader comments posted under news articles. The EMBEDDIA project focuses on the use of cross-lingual techniques to transfer language technology resources to less-resourced languages (as well as English and Russian, the project focuses on Slovene, Croatian, Estonian, Lithuanian, Latvian, Finnish, and Swedish), and the application of these to real-world problems in the news media industry. One such problem is the need for news publishers to allow readers to post comments under articles online, in order to promote engagement with the content, but prevent content being published that would be offensive to other readers, dangerous or in some way compromise the legal position of the publisher. Most publishers currently use manual methods to do this: a team of moderators will monitor comments and block them when required. However, high volumes of comments can often make this impractical. The use of automatic natural language processing methods to detect comments that should be blocked, or referred to human moderators, can speed up the process many times (Pavlopoulos et al., 2017a); and many successful approaches to automated hate and offensive speech detection and categorisation exist (see e.g. MacAvaney et al., 2019; Schmidt & Wiegand, 2017), with datasets and shared tasks made available for several major EU languages (see e.g. Zampieri et al., 2019; V. Basile et al., 2019). However, such resources are generally only available for a few languages (e.g., English, German), leaving a gap for less-resourced languages. For Estonian and Croatian, languages of interest here, the number of studies is very limited (Ljubešić et al., 2018).

In this work, we describe new data collection efforts in two less-resourced European languages (Croatian, Estonian), and our experiments into automated classification. We explain the existing moderation scheme used by humans in news editorial houses, and examine to what extent it

[1] http://embeddia.eu

overlaps with the concept of offensive language as usually defined; describe a range of suitable classifier architectures for automatic detection of problematic comments; and give results showing that although reasonable performance can be achieved on these languages given suitable methods, it does not reach the levels that might be expected given other related work in languages in which more resources are available. We then examine the robustness of both the classifiers and the moderation scheme itself, and find that performance is limited not only by the nature of interactive language and its dependence on context, but by the need to rely on labels gathered under real-world constraints. We conclude that a transfer learning approach is the most promising future direction, providing the opportunity to incorporate information from more, better-curated datasets available in other languages, but that this will require cross-lingual techniques beyond the current state of the art.

2. Data and Task

The task of interest here, broadly defined, is to develop an automatic classifier to automate (or partially automate) the manual process of moderation: deciding which reader comments should be blocked, according to the policy of a particular newspaper.

2.1. Dataset

For this work, we have collected a large new dataset of online reader comments, from a range of news media sources in two less-resourced European languages, as covered by our project partners. Our dataset consists of over 60 million comments from the articles published online by three major news outlets:

- **24sata (`www.24sata.hr`):** The largest-circulation daily newspaper in Croatia, reaching on average 2 million readers daily.[2] Language: Croatian. Size: 21.5M comments.
- **Večernji List (`www.vecernji.hr`):** The third-largest daily newspaper in Croatia. Language: Croatian. Size: 9.6M comments.
- **Eesti Ekspress (`www.ekspress.ee`):** The largest weekly newspaper in Estonia, with a circulation of over 20,000. Languages: Estonian, Russian (articles are written in Estonian, but comments are often also in Russian). Size: 31.5M comments.

2.2. Annotation

In each case, the comments are annotated with metadata including link to the relevant article, ID of the comment author (anonymised) and timestamp; importantly for the purposes of this work, comments are also labelled if they are blocked by human moderators. Details of the moderation policy, and therefore the nature of the labelling, vary with news source, but comments may be blocked for a wide range of reasons. For 24sata, the annotation reflects a moderation policy based on 8 different categories, shown in Table 1; comments should be blocked if they breach any one of

[2] `https://showcase.24sata.hr/2019_hosted_creatives/medijske-navike-hr-2019.pdf`

these categories, although the implications for the comment author vary with the severity of the category. Less serious offences (labelled ‘minor’ in Table 1) lead to a minor warning: a user may receive up to two minor warnings, but the third one leads to a temporary one-day ban from the site. More serious offences lead to major warnings, of which a user may only receive one – the second one leads to a five-day ban. After a ban, the number of warnings of that type are reset to zero, but breaking the rules multiple times can, at the discretion of the moderators, lead to a permanent ban.

Rule ID	Description	Definition	Severity
1	Disallowed content	Advertising, content unrelated to the topic, spam, copyright infringement, citation of abusive comments or any other comments that are not allowed on the portal	Minor
2	Threats	Direct threats to other users, journalists, admins or subjects of articles, which may also result in criminal prosecution	Major
3	Hate speech	Verbal abuse, derogation and verbal attack based on national, racial, sexual or religious affiliation, hate speech and incitement	Major
4	Obscenity	Collecting and publishing personal information, uploading, distributing or publishing pornographic, obscene, immoral or illegal content and using a vulgar or offensive nickname that contains the name and surname of others	Major
5	Deception & trolling	Publishing false information for the purpose of deception or slander, and “trolling” - deliberately provoking other commentators	Minor
6	Vulgarity	Use of bad language, unless they are used as a stylistic expression, or are not addressed directly to someone	Minor
7	Language	Writing in other language besides Croatian, in other scripts besides Latin, or writing with all caps	Minor
8	Abuse	Verbally abusing of other users and their comments, article authors, and direct or indirect article subjects, calling the admins out or arguing with them in any way	Minor

Table 1: Annotation schema for blocked comments, 24sata.

As Table 1 shows, the categories cover a broad range of grounds for moderation, and many categories potentially overlap. They include a range of categories in the broad area of offensive language, many of which might overlap: threats to others (rule 2); hate speech based on national, racial, sexual or religious affiliation (3); obscene or immoral content (4); bad language (6); and verbal abuse (8). However, they also include a range of other reasons: illegal content (rule 1); comments not allowed by the portal’s rules (1); advertising (1); off-topic posts (1); copyright infringement (1); false information (5); use of language other than Croatian (7).

Rule ID	Corresponding 24sata rule ID(s)	Definition	Severity
1	3	Hate speech on a national, religious, sexual or any other basis	Major
2	2	Threats to other users, administrators, journalists or subjects of articles	Major
3	6, part 4, part 8	Insulting other users or use of bad language.	Minor
4	part 4	Publishing personal data	Minor
5	part 1, part 7	Chat, off-topic, writing in all caps, posting links	Minor
6	part 7	Writing in a script other than a Latin script	Minor
7	part 8	Challenging the administrators or arguing with then in any way	Minor
8	part 5	Posting false information	Minor
9	n/a	Using multiple user accounts	Permanent ban

Table 2: Annotation schema for blocked comments, Večernji List, together with corresponding Rule IDs from the 24sata schema (Table 1).

Furthermore, as Table 2 shows, these categories also vary between publishers: the categories for Večernji List (hereafter VL) have many similarities with those for 24sata, but it is not possible to map directly between them. Categories such as hate speech and threats seem to correspond directly (rules 3 and 2 for 24sata, rules 1 and 2 for VL); but others are combined in different ways (e.g. 24sata's rule 5 covers posting false information, which maps to VL's rule 8, but also covers trolling and povocation which does not seem to be explicitly covered in VL's policy; VL's rule 3 covers insults and bad language, aspects of which are covered by parts of 24sata's rules 4, 6 and 8). Ekspress, on the other hand, do not record explicit categories of policy violation, so no such detailed annotation is available.

Three distinct problems therefore arise. First, distinguishing between the categories — rather than just detecting the general category of requiring moderation — is an important task in order to record how the policy was applied when blocking a comment or banning a user, where such a policy exists. Second, the overall category of blocked comments is likely to cover a very heterogeneous sample of language, as it results from a diverse range of phenomena. Third, as the categories are not *a priori* fixed, and can be conceptually divided up in different ways, this heterogeneity is likely to extend even to the individual classes.

Problematic comments are fairly common: for the 24sata subset, articles receive around 45 comments on average, and those that receive problematic comments receive around 5.5 of them. However, the data is highly unbalanced — only around 5-6% of comments require blocking — bringing an added complication to the classification task.

3. Related Work and Resources

In this section, we investigate what resources might be available which can help; in particular, what datasets might be available to provide training data for suitable classifiers.

3.1. Comment Filtering

Previous work in news comment filtering is limited. Pavlopoulos et al. (2017a) address the problem using data from a Greek newspaper, Gazzetta. They use a dataset of 1.6M comments with labels derived from the newspaper's human moderators and journalists; they test a range of neural network-based classifiers and achieve encouraging performance with AUC scores (area under the ROC curve) of 0.75-0.85 depending on the data subset. However, being in a different language (Greek) their data is not directly usable as a training set for our task. In addition, their moderation labels are binary, representing a "block or not" decision, rather than giving any further information about the reasons behind a decision. They are therefore not suited to investigating the moderation policy labels of interest here; and more fundamentally, it is unclear whether the decisions of Gazzetta's moderators are based on similar aims or policies as the decisions we must try to simulate for 24sata or Ekspress's moderators. Pavlopoulos et al. (2017a) asked additional annotators to classify comments according to a more detailed taxonomy (*"We also asked the annotators to classify each snippet into one of the following categories: calumniation (e.g., false accusations), discrimination (e.g., racism), disrespect (e.g., looking down at a profession), hooliganism (e.g., calling for violence), insult (e.g., making fun of appearance), irony, swearing, threat, other."*) but this was done as a post-hoc exercise and only for a small portion of the test set. It was not used in classification experiments, but only for separate analysis purposes.

Other work with reader comments on news (see Table 3) exists but does not attempt to learn from or reproduce moderation decisions directly in the same way. Kolhatkar et al. (2019) and Napoles et al. (2017) investigate constructivity in comments, and provide datasets which distinguish between constructive and non-constructive comments; these datasets are related to our task, though, as they also include information about toxicity and related categories such as insults and off-topic posting. Barker et al. (2016) investigate quality of comments and their use in summarisation. Wulczyn et al. (2017) investigate a related problem of detection of personal attacks and toxicity in user comments on Wikipedia articles, rather than news; and Zhang et al. (2018) also investigate Wikipedia comments from the point of view of detecting which conversations become toxic. None of these directly solve our problem, although they could in theory provide useful information; however, all are limited to English data.

3.2. Resources for Related Tasks

A variety of related tasks have been studied in data other than user-generated comments on articles. Given the moderation policy details in Section 2 above, the existence of suitable datasets for training classifiers for various categories of offensive language, advertising/spam, and trolling behaviour would be of interest. While none of these categories corresponds directly to the overall category of comments that must be blocked, each one covers a phenomenon that requires blocking.

Corpus	Location	Domain	Language	Size	Type of annotation
Gazzetta	(Pavlopoulos et al., 2017a)	News	gr	1.6M	Moderation
SFU SOCC	(Kolhatkar et al., 2019)	News	en	663k	Constructiveness, toxicity
YNACC	(Napoles et al., 2017)	News	en	522k	Constructiveness, insults, off-topic
SENSEI	(Barker et al., 2016)	News	en	2k	Quality, tone, summaries
DETOX	(Wulczyn et al., 2017)	Wiki	en	115k	Personal attacks, aggression, toxicity
Zhang et al., 2018	(Zhang et al., 2018)	Wiki	en	7k	Personal attacks

Table 3: Existing datasets for filtering user-generated comments on articles. Size is given in number of comments.

Corpus	Location	Domain	Language	Type of annotation
FRENK	(Ljubešić et al., 2019)	Facebook	en,sl	Socially unacceptable language
HASOC	hasoc2019.github.io	Twitter/Facebook	de, en, hi	Hate speech, target
HatEval 2019	(V. Basile et al., 2019)	Twitter	en, es	Hate speech, target, aggression
OLID (OffensEval)	(Zampieri et al., 2019)	Twitter	en	Hate speech, target, threats
GermEval	(Wiegand et al., 2018)	Twitter	de	Abuse, profanity, insults
IBEREVAL	(Anzovino et al., 2018)	Twitter	en,es	Misogynous
MEX-A3T	(Álvarez-Carmona et al., 2018)	Twitter	es-mx	Aggressive
Liu et al 2018	(Liu et al., 2018)	Instagram	en	Hostile
Waseem & Hovy 2016	(Waseem & Hovy, 2016)	Twitter	en	Hate speech, with subcategory
Stormfront	(de Gibert et al., 2018)	Online forum	en	White supremacy

Table 4: Existing datasets: abuse, hate speech and offensive language. “Target” refers to annotation of the group or individual towards which hate speech is directed.

3.2.1. Offensive Language Detection

Recent years have seen a large amount of research on detection of offensive language of various kinds. Many public datasets have been created and distributed, many shared tasks have been run, and many classification systems developed and tested (see Table 4). The exact definition of the categories annotated in these tasks varies, however (see Schmidt & Wiegand, 2017, for a survey), and may include one or all of:

- Threats: hostile speech intended to threaten the addressee with violence or other negative effects;
- Abuse: personal insults directed at others, including ‘flaming’ or cyberbullying;
- Hate speech: personal attacks on the basis of religion, race, sex, sexuality etc.;
- Offensive content: the use of language which is in itself considered rude, vulgar or profane (including pornographic), even if not targeted at someone in particular.

These terms are often used interchangeably, with some (particularly *hate speech*) often used to cover multiple categories. Exact definitions of the individual categories also vary with task and dataset, so we do not attempt an exhaustive exposition here. As an illustrative example, Waseem & Hovy (2016) define their *hate speech* category for Twitter as a message that:

1. *uses a sexist or racial slur;*
2. *attacks a minority;*
3. *seeks to silence a minority;*
4. *criticizes a minority (without a well founded argument);*
5. *promotes, but does not directly use, hatespeech or violent crime;*
6. *criticizes a minority and uses a straw man argument;*
7. *blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims;*
8. *shows support of problematic hash tags. E.g.“#BanIslam”, “#whoriental”, “#whitegenocide”;*
9. *negatively stereotypes a minority;*
10. *defends xenophobia or sexism;*
11. *contains a screen name that is offensive, as per the previous criteria, the tweet is ambiguous (at best), and the tweet is on a topic that satisfies any of the above criteria.*

On the other hand, Ljubešić et al. (2019) use a more restrictive set of definitions via a decision tree to separate out different kinds of *socially unacceptable discourse (SUD)* on Facebook into different categories:

Is this SUD aimed at someone’s background?

- YES: Are there elements of violence?
  - YES: background, violence
  - NO: background, offensive speech
- NO: Is this SUD aimed towards individuals or other groups?
  - YES: Are there elements of violence?
    - YES: other, threat
    - NO: other, offensive speech
  - NO: Is the speech unacceptable?
    - YES: inappropriate speech
    - NO: acceptable speech

In all these variants, the task is usually defined as a classification task — detecting whether a given text should be classified as hate speech (or abuse, offensive language etc.) or not — although this may be set up as a binary or a multi-class classification problem depending on the definitions used. Many datasets are available for this broad category of tasks, with a number of public shared tasks having been run over the last few years.[3] The exact categories annotated vary, as do the domain and language of text annotated; we give an indication of each in Table 4.

Most datasets are based on social media (mainly Twitter) posts. Performance varies widely with dataset and domain. OffensEval 2019 reports maximum F1 score 0.829 on the offense classification task; for the white supremacy forum comments (de Gibert et al., 2018) classification accuracy is 0.78.

3.2.2. Spam detection

Another important task for UGC filtering in many domains, corresponding to one of the categories in the 24sata moderation policy in Section 2, is the detection of *spam*: comments which are off-topic, intended not to contribute to an ongoing conversation or relate to a given topic but rather to advertise, and/or to entice readers into clicking on a link either to generate revenue or for more nefarious purposes (e.g. 'phishing', attempting to gain access to personal information). This task is highly relevant for news media companies in order to prevent comments sections being taken over by irrelevant, offputting or dangerous content.

The task is a variant of the familiar spam detection problem for email (see Caruana & Li, 2012, for a survey), but UGC and online comments have their own distinctive characteristics – see for example (Kantchelian et al., 2012) for application to comments in the blog domain, (Aiyar & Shetty, 2018) in the Youtube domain, and (Wu et al., 2018) for a survey of work in the Twitter domain.

Corpus	Location	Size	Language	Domain
NSC Twitter Spam	(Chen et al., 2015)	6 million tweets	en	Twitter
Youtube Spam Collection	(Alberto et al., 2015)	1956 comments	en	Youtube
MPI-SWS	(Ghosh et al., 2012)	41,352 accounts	n/a	Twitter

Table 5: Existing datasets: spam.

Corpus	Location	Size	Language	Domain
FiveThirtyEight	(Linvill & Warren, 2018)	2,973,371 tweets	en	Twitter
Dataturks	(Narayanan, 2018)	20,000 tweets	en	Social media
Mojica 2017	(Mojica de la Vega & Ng, 2018)	5,868 conversations	en	Reddit

Table 6: Existing datasets: trolling and incitement.

Table 5 shows a sample of the most relevant datasets here. Alberto et al. (2015) provide a dataset of comments on Youtube videos classified as spam or not. Several datasets are available for short text messages in social media, see e.g. (Chen et al., 2015)'s large collection of 6 million spam tweets, and the MPI collection of Twitter accounts detected as spam accounts. Again, this task is usually defined as a binary classification task. Performance varies widely with dataset and

[3] A helpful catalogue of relevant datasets is also available online at `http://hatespeechdata.com/`.

domain. Wu et al. (2018) report accuracies of up to 94.5% on account classification and 88-91% accuracy on individual texts.

3.2.3. Trolling and incitement

Another basis for moderation in the policy of Section 2 is the presence of *trolls* and *bots*: users who may be automated or semi-automated rather than human, and which behave in a disruptive and/or deceptive manner in order to influence discussion, spread propaganda and manipulate opinion or to incite extreme views and disrupt discussion (see e.g. Kim et al., 2019). The effects of such agents in social media and news article comments can be strong, with evidence that they have affected public opinion and outcomes of elections (Badawy et al., 2018). There is a connection with the *fake news* phenomenon, with many trolling accounts being used to spread false rumours and link to fake news.

In this case, although this can be approached in a similar classification manner to the tasks above, labelling texts as coming from trolls, the problem is more often seen as one of classifying user accounts rather than their individual text outputs. Methods used therefore often depend as much on the social network properties of user accounts as on the language they generate. Again, some datasets exist; see Table 5. FiveThirtyEight distribute a dataset of nearly 3 million tweets sent from Twitter accounts *"connected to the Internet Research Agency, a Russian "troll factory" and a defendant in an indictment filed by the Justice Department in February 2018"* between February 2012 and May 2018. Narayanan (2018) then provides a smaller dataset from the same source, but annotated in more detail for level of aggression. Mojica (2017); Mojica de la Vega & Ng (2018) collected a similar dataset of comments on Reddit.

In our domain of UGC comments under news articles, Mihaylov & Nakov (2016) collected a dataset from over 2 years of articles (Jan 2013-April 2015) on the Bulgarian news site Dnevnik (`dnevnik.bg`), totalling 1,930,818 comments by 14,598 users on 34,514 articles. Troll comments were identified by a combination of observing other users' reactions, and checking identities in leaked documents; however, the dataset is not currently available publicly.

Mihaylov & Nakov (2016) achieve around 81% accuracy and F-score on the classification task, on a balanced dataset of news comments, using simple baseline linear classifiers. Mojica (2017) achieves c.90% accuracy on his dataset for the trolling detection task, using a more complex conditional random field classifier.

3.3. The Problem of Monolinguality

As the discussion above shows, datasets are available. However, very few are in the exact domain of automatic moderation: the Gazzetta dataset of (Pavlopoulos et al., 2017b) is the only example from news, with the Wikipedia dataset of (Wulczyn et al., 2017) being quite closely related. More critically, none are available in the languages required here (Croatian, Estonian); the closest are the Facebook dataset of socially unacceptable discourse in Slovenian of Ljubešić et al. (2019), and the Bulgarian news comment trolling data of Mihaylov & Nakov (2016), but neither are publicly available and neither are in the exact domain required.

This problem is a widespread one in NLP: a large majority of research and available datasets is monolingual and in English, and datasets for specific less-resourced languages like Croatian and Estonian are hard to find. Some multi-lingual work exists: Ousidhoum et al. (2019) present a multilingual hate speech study on English, French and Arabic tweets, and A. Basile & Rubagotti (2018) conduct cross-lingual experiments between Italian and English; again, this does not cover our languages or domain.

We also note the existence of Hatebase,[4] a highly multilingual collection of crowdsourced social media posts; however, as its annotation is based only on submission by the public, and it contains no comparable non-abuse language, it is not currently suitable as training or evaluation data for a classifier of the kind needed here.

We therefore conclude that for our present purposes, training on the specific data we have, in the correct language and reflecting the moderation policy of the correct newspaper, is the only practical option. The next section outlines our experiments using this approach.

4. Experiments

Our approach is therefore to treat the task as a classification problem, and use the real-world moderator decisions, recorded in the newspaper databases, as our training and test labels.

4.1. Classification Models

We formulate the problem as a text classification task. The basic task is a binary choice: given a comment, a system has to predict whether it should be *blocked* or *non-blocked*. We can also consider a multi-class task: given a comment, to predict which rule (Table 1 or Table 2) is being violated. We compared four different models, each using a standard method for text classification.

Naïve Bayes As a baseline, we use a Naïve Bayes (NB) classifier. NB is a simple probabilistic generative model which makes the approximation that words are independent of one another: the probability of a text belonging to a particular class can therefore be approximated as the product of the probabilities of the individual constituent words being associated with that class, and those can be calculated directly from frequencies in the training set. While clearly an oversimplification, this approach can provide good results in many text classification tasks, including spam detection (see e.g. Jurafsky & Martin, 2009). It also provides an easily interpretable model: a conditional probability table relating each word to each class.

LSTM In this model, the comment is encoded using a Long Short-Term Memory (LSTM) recurrent neural network (Hochreiter & Schmidhuber, 2015): LSTMs are able to encode not only word sequence but capture dependencies between non-adjacent words. The last hidden state of the LSTM is taken as the representation of the comment, and on top of that, a multi-layer perceptron (MLP) is used to produce the classification decision. Word embedding vectors are randomly initialised, and the whole architecture is trained end-to-end.

[4] `http://hatebase.org/`

LASER In this model, the comment is represented using Language-Agnostic SEntence Representation (LASER, Artetxe & Schwenk, 2019). LASER produces representations for sentence-length texts, obtained using a five-layer bidirectional LSTM (BiLSTM) encoder with a shared byte-pair encoded (BPE) dictionary for 92 languages. The last states of the LSTM are used to produce a sentence vector by max-pooling, and the model is trained using an encoder-decoder approach, in which the sentence representations are used to generate parallel sentences in another language. This approach gives sentence vectors which capture many aspects of sentence meaning and can be used in many tasks; here, we use a MLP on top of the sentence representations, and train it on our classification task. Only the MLP is trained; the weights of the LASER encoder are kept frozen using the pre-trained models available.[5]

mBERT In our final model, the comments are represented using Bidirectional Encoder Representations from Transformers (BERT, Devlin et al., 2018). BERT is a deep contextual representation based on a series of layers of Transformer cells (Vaswani et al., 2017), and trained using a variant of a language model objective. As with LASER above, we then pass the comment representation to a MLP for classification. The BERT model weights are initialized using the multilingual pre-trained model (mBERT, trained on 104 languages by sharing embeddings across languages), and fine-tuned end-to-end along with the MLP.[6]

Training Note the difference in the training strategy for our LSTM, LASER, and mBERT models. In the case of LSTM, the whole architecture is initialized randomly and trained end-to-end: we use no pre-trained embeddings, and train only on the data available here. In the case of LASER, only the classification MLP weights are trained, while the LASER model sentence (comment) representation weights are kept fixed at the values in the pre-trained model. For mBERT, the comment representation weights are initialized using the pre-trained model, and the MLP weights initialized randomly, and the whole model is then fine-tuned end-to-end. All the neural models are trained using the Adam optimizer (Kingma & Ba, 2014) with cross-entropy loss.

4.2. Experiment 1: Binary Classification

4.2.1. Data Selection

As Figure 1 shows, the rate of commenting on articles, and the rate at which moderators block comments, vary over time. (Detailed frequency counts are given in Appendix A, Section A.1). For Ekspress, the rate of commenting rises steadily over time; for 24sata, it rises to a peak in 2015/2016 and then reduces slightly. For VL, the commenting rate seems more stable. (Note that the data was collected part-way through the year 2019, so data for that year is not for a complete year period). Particularly of note, though, is that the rate at which moderators block comments rises over time for all newspapers; the effect is particularly marked for VL from 2013 onwards, and for 24sata from 2016 onwards. Note that the rates for VL before 2013, and 24sata before 2016, are not zero, but very low; see Appendix A for details. This effect is not merely one of

[5] Pre-trained model available from `https://github.com/facebookresearch/LASER`.

[6] Pre-trained model available from `https://github.com/google-research/bert`.

comment volume: higher commenting rates do not correspond to higher blocking rates (Figure 1), as might be hypothesized if, say, a rise in commenting rates were caused by a sudden influx of troll accounts or an increase in contentious topics. Instead, the most likely cause is a change in moderation policy: over recent years, more attention has been given by newspapers to moderation, in terms of both overall importance and strictness of adherence to policy. Note also that blocking rates are relatively low in general: even the peak rate for VL is only just over 15% of comments, for Ekspress 12.5%, and for 24sata only 7.8%: this gives an unbalanced dataset which must be accounted for in training and testing.

Given the sharp change over time, it seems very likely that data from more recent years will be more consistent, and will be more reflective of current moderation policy: earlier years are likely to contain large numbers of false negatives (comments that were not moderated at the time, due to either lack of resources or difference in policy, but would be blocked now). In order to have the cleanest and most relevant data possible, we therefore first selected 2019 data for training, validation, and testing purposes. Since most comments are non-blocked comments, to have a balanced dataset for experiment purposes, we first selected only those articles which have at least one blocked comment. We then divided those articles into training (80%), validation (10%) and test (10%) partitions. Finally, we randomly selected an equal number of blocked and unblocked comments per article in each set. Table 7 shows the resulting data distribution for all three newspapers.

	24sata			Večernji List			Ekspress		
	Train	Val	Test	Train	Val	Test	Train	Val	Test
# Articles	9196	1148	1154	6521	813	821	7490	934	942
# Comments	99246	12364	12472	85916	10490	10855	145154	19310	20312

Table 7: Partitioned dataset distribution, 24sata, Večernji List and Ekspress.

4.2.2. Results

Table 8 shows the results for each classifier model. As our training and test sets have an evenly weighted number of positive (blocked) and negative (non-blocked) examples, we give performance as standard percentage accuracy, and to get an insight into the relative performance we give this not only overall but over the positive and negative portions of the test set individually. ‘Blocked’ accuracy is therefore equivalent to recall for the positive (blocked) class; ’Non-blocked’ accuracy is recall for the negative (non-blocked) class. Standard summary measures such as weighted average F-score are not very helpful in this setting, as they can be so strongly dominated by the majority (non-blocked) class, and accuracy on the two classes has different implications for news publishers; we therefore examine per-class metrics (although see Section 4.4 for results in terms of macro-averaged F-score on the final dataset).

For all three newspapers, the mBERT model gives best performance. Surprisingly, the NB model gives relatively strong performance, with neither the LSTM nor LASER models providing much of an improvement; in fact, for Ekspress they perform worse than NB. Accuracy is higher for 24sata than for Ekspress and VL, but in all cases the absolute level of accuracy is lower than might

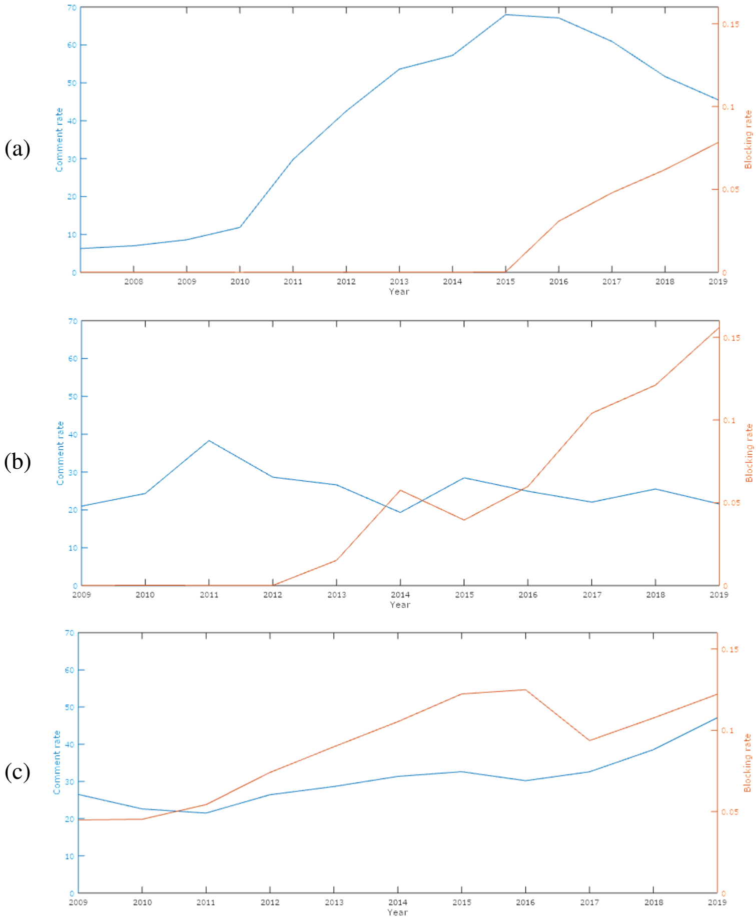


Figure 1: Comment rate $N_{\text{comments}}/N_{\text{articles}}$ in blue, and blocking rate $N_{\text{blocked}}/N_{\text{comments}}$ in red, over time, for (a) 24sata, (b) Večernji List, (c) Ekspress.

be expected given comparable experiments with offensive language detection in other research (Section 3). Accuracy on blocked content is lower than the accuracy of recognition of non-blocked content, particularly for Ekspress.

To calculate the performance that would be expected on real (unbalanced) data, we must take into account the expected real ratio of blocked to non-blocked comments. As Section 2 discusses, blocked comments are rarer than non-blocked, with the most recent estimate of the ratio from 2019 being 0.078 for 24sata. In practice, we would therefore expect for 24sata a recall of 0.67, a precision of 0.27 and an F-score of 0.38. In other words, the classifier would successfully detect 67% of comments that needed blocking (missing 33%), but 73% of its decisions to block would be false positives; and nearly 15% of innocent comments would be falsely blocked. While this level of performance is potentially useful, it seems it would still require significant manual filtering on the part of moderators. The balance between recall and precision could of course be tuned via the decision boundary, or by weighting the objective function in training, but gains in the recall would correspond to losses in precision, and vice versa (see Pavlopoulos et al., 2017a).

	24sata			Večernji List			Ekspress		
Model	ALL	BLK	NON	ALL	BLK	NON	ALL	BLK	NON
NB	69.43	47.59	**91.26**	66.39	49.75	81.79	64.57	46.48	82.66
LSTM	71.52	61.70	81.33	65.39	54.47	75.50	63.02	41.96	**84.09**
LASER	70.74	**70.11**	71.36	63.31	**59.77**	66.59	61.58	47.07	76.10
mBERT	**76.42**	67.33	85.49	**69.63**	53.18	**84.87**	**68.40**	**58.46**	78.34

Table 8: Classifier performance, as percentage accuracy. Columns are labelled ALL for all comments, BLK for positive instances only (blocked content), NON for negative instances only (non-blocked content).

Inspection of the conditional probability table produced by the NB model allows us to determine the words which are most strongly associated with the blocked and non-blocked classes, on the basis of the ratio of class probabilities. Tables 21 and 22 in Appendix B show full lists of the top 100 words for each class for 24sata. The strongest indicators for the blocked class correspond to vocabulary expected in spam comments: external URLs (*www*, *com*, *google*, *posjetite* (visit)); work and earnings (*poslu/posla* (work), *plaća* (payment), *zaradio/zaraditi* (earn)); amounts of money promised (numbers, *dolara* (dollars), *eura* (euros), *mjesecu* (monthly), *tjedno* (weekly), *dnevno* (daily)). Vocabulary associated with offensive language is also included, but comes further down the list (*jebem/jebo* (fuck), *majmun* (monkey)). Non-blocked indicators include vocabulary associated with discussion of a range of news topics (e.g. football: *inter*, *derbi*) and general evaluative words (*sretno/sritno* (happy/good luck), *predivno* (amazing), *najljepša* (most beautiful), *strašno* (terrible)). However, of a list of 185 blacklisted words used by the moderators at 24sata to flag comments for blocking, only 78 appear in the top 1000 in the NB model; and surprisingly, many words that one might expect to be associated with offensive or highly-charged language (although no blacklisted words) appear in the top 1000 non-blocked indicators in the NB model: *svastiku* (swastika), *terorizam* (terrorism), *trolaš* (you're trolling).

Vocabulary indicators extracted from these annotations are therefore not straightforward, suggesting that the data is fairly heterogeneous: comments may be blocked for many diverse reasons, and therefore display very different textual features. This may be one possible reason for the below-par performance; our next experiment investigates this.

4.3. Experiment 2: Blocking Rule Classification

For 24sata and VL, the publisher's database records the reason behind the moderators' decisions: the specific rule that a comment breaks. Here, we train and test multi-class versions of our classifier models for the problem of rule recognition.

(a)

	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
Train	24329	20	2167	30	2912	992	387	18786
Val	3081	1	216	1	271	114	41	2457
Test	2962	1	248	2	388	134	57	2444

(b)

	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8	Rule9
Train	3652	6548	4514	57	3756	9	156	4	24322
Val	572	794	547	7	402	0	13	0	2914
Test	553	864	580	4	456	2	24	0	2951

Table 9: Blocking rule dataset distribution, for (a) 24sata and (b) Večernji List.

Table 9 shows the distribution of blocked comments by rule within the training, validation and test sets defined above. The distribution is very uneven: for 24sata, rules 1 (unrelated topics, spam, advertising etc.) and 8 (abuse, arguing with administrators) are common, while rules 2 (direct threats) and 4 (obscenity) are extremely rare; others are in between. For VL, rule 9 (using multiple accounts) is most common, with rules 4 (publishing personal data), 6 (using non-Latin script) and 8 (misinformation) very rare. Even for rules which seemingly map directly between the two schemata (e.g. hate speech: 24sata rule 3, VL rule 1; threats: 24sata ule 2, VL rule 2) the distributions seem to vary widely across newspapers: it seems to be very rare for 24sata moderators to class comments as threats, but quite common in VL.

One hypothesis might be that moderators tend to avoid applying rules with more serious consequences if other less serious ones could be used (see Tables 1 and 2); but while this might explain the rarity in 24sata of rules 2 (threats) and 4 (obscenity), it does not explain the distribution in VL, where rules 1 (hate speech), 2 (threats) and 9 (multiple accounts) are all commonly used. It may be that the ambiguity of many rules, together with the cultural practices and habits within particular groups of moderators, have significant effects here.

Results Table 10 shows the results for individual rules, with Table 11 showing the effect this would have on overall blocking accuracy (comments which break any rule should be blocked).

Performance for individual rules varies widely. Less frequent rules are often ignored by all classifiers (rules 2, 4), with better performance for more frequent rules (e.g. rules 1, 8). It is likely that the lower contribution of the less frequent classes to the training objective function means that not enough weight is given to them in the final classifier models. The NB model does much worse than other models, presumably because the pruning of the conditional probability table favours more common words, likely to be significant indicators of the more common classes. The simpler LSTM model seems to have an advantage over the more complex LASER and BERT models, in that accuracy seems more even across classes; this may be because the pre-training of the LASER and BERT models gives them less ability to adjust to the different classes in fine-tuning.

However, the overall performance is not strongly affected. Given the real blocking rate, for 24sata we would expect a recall of 0.48, a precision of 0.32 and an F-score of 0.39. This translates to successfully detecting 48% of comments that needed blocking (missing 52%), while producing 68% false positives; and blocking nearly 8% of innocent comments. Note that the F-score is very

	Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
(a)	NB	43.01	0	3.23	0	5.67	5.97	8.77	2.74
	LSTM	62.42	0	56.05	0	50.52	75.37	43.86	57.53
	LASER	51.25	0	9.68	0	1.55	16.42	0	50.12
	mBERT	48.68	0	0	0	0	0	0	63.3

	Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8	Rule9
(b)	NB	6.61	5.47	4.56	0	6.4	100	4.55	0	33.73
	LSTM	25.73	20.64	33.65	50	35.22	0	13.64	0	40.41
	LASER	51.39	45.26	67.49	66.67	61.37	0	63.64	0	57.85
	mBERT	0	0	43.54	0	0	0	0	0	42.01

Table 10: Blocking rule classifier performance, measured as percentage accuracy, (a) 24sata (b) Večernji List.

Model	Overall	Blocked	Non-blocked
Chance	11.11	11.11	11.11
NB	60.06	22.19	**97.93**
LSTM	**71.78**	**59.59**	84.16
LASER	67.09	44.82	89.35
mBERT	70.04	47.93	92.19

Table 11: Performance of multi-class rule classifier on binary task, measured as percentage accuracy, 24sata.

similar to the classifier trained on the binary task, although the balance between precision and recall is different; this could be adjusted as discussed above.

To investigate the role of the multi-class objective function in training, we also checked the coverage of the classifiers trained on the binary task in Section 4.2 above. While these classifiers give only binary output and therefore cannot help moderators understand decisions, we can compare their ability to detect the individual rules. Table 12 shows the results. The very rarest classes (rules 2, 4) seem to behave quite randomly (given the very low counts, this is not surprising), but the slightly more common rules (6 and 7, then 3 and 5) get reasonable accuracy for most classifiers. The picture is mixed, however: some classes seem to be inherently hard to detect, with rules 5 (trolling) and 7 (non-Croatian language) getting relatively low scores for all classifiers.

Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
NB	52.77	0	45.56	0	27.84	71.64	22.81	43.99
LSTM	63.37	100	61.29	50	52.84	79.85	56.14	60.27
LASER	71.0	100	69.76	100	58.25	84.33	42.11	70.79
mBERT	64.15	0	72.18	100	54.64	88.06	36.84	72.3

Table 12: Performance of binary classifier per blocking rule, measured as percentage accuracy, 24sata.

4.4. Experiment 3: Variation over Time

Another possible reason for variable performance is the reliability and/or variability of the moderation annotation itself. Moderation can be quite a subjective decision, and the large amounts of data to mean that many blockable comments may be missed. One way to test this is to examine how classifier performance changes over time, as the moderation policy and the amount of effort put into moderation changed over the years (see Section 4.2.1); for this experiment we focus on just one dataset, 24sata. The distribution of individual blocking rules also varies over time: Figure 2 shows the proportion of blocking decisions based on each rule for the last four years (the years with most data). (Full details of the rule distributions over time for both 24sata and VL are given in Appendix A, Section A.2). Significant changes can be seen in the proportions. Some changes may reflect changes in behaviour: for example, rule 1 (advertising/spam) is used progressively more over time. However, other changes may be more complex: the commonly used 8 (abuse) becomes less used over time, with related rarer classes such as 2 (threats) and 5 (trolling/provocation) increasing. It therefore seems likely that rules are being applied differently in different cases: with many rules covering a range of phenomena and many phenomena being covered by multiple rules (see details of the rules in Table 1 above), moderators have a choice in which rules to apply, and perhaps more specific rules (often with more stringent penalties) are becoming preferred.

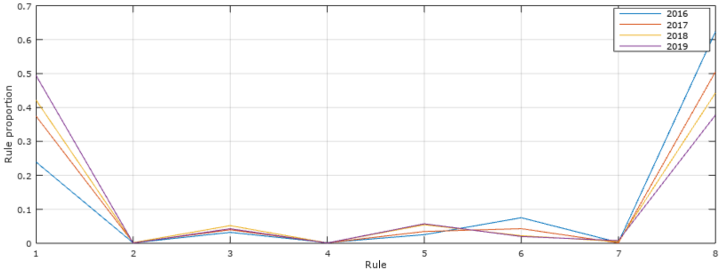


Figure 2: Blocking rule proportion over time, 24sata.

To determine the variability of the models' performance over different years' data, we therefore created a series of test sets, one for each of the last four years. We keep the same training set, taken from 2019 data (see above); the 2019 test set is therefore smaller and based on that used in the previous section. The test sets for 2016-2018 are larger as they can contain all the year's data labelled with rules; as the training set is fixed we can also test on a realistic balance of data, using all the blocked and non-blocked comments available for each year. Table 13 shows the test set distribution over time.

Results Table 14 shows overall accuracy figures per year on the 24sata dataset; we show only performance for the best classifier model, mBERT. Accuracy decreases as we move further away from the year 2019 used in training. Table 15 then shows how the accuracy of the binary blocking

	Articles	Non-blocked	Blocked	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
2016	907	196762	15154	2915	111	992	183	683	1413	227	8630
2017	1045	188639	20579	6351	185	1560	153	1273	1211	137	9709
2018	1678	285620	21838	237	254	2800	125	2616	840	780	14186
2019	1154	68706	6398	3070	3	256	2	396	138	58	2475

Table 13: Yearwise dataset distribution, 24sata.

Year	Overall	Blocked	Non-blocked	F1-macro	Recall (BLK)	Precision (BLK)
2016	72.25	72.20	72.89	54.19	0.73	0.15
2017	75.17	76.16	64.84	58.10	0.65	0.21
2018	76.75	78.36	61.32	59.59	0.61	0.23
2019	80.03	81.19	67.32	62.07	0.67	0.25

Table 14: Binary classification performance over the yearwise testset using mBERT, 24sata. Figures are shown as percentage accuracy overall and for the blocked and non-blocked content separately; as this experiment uses the full data for each year (rather than a balanced subset) we also give F1 score macro-averaged over the two classes, and recall and precision for the blocked class only.

classifier varies with blocking rule class: while figures for many rules decrease in years before 2019, performance for rules 3 (hate speech), 6 (vulgarity) and perhaps 8 (abuse of other users, authors and admins) seems to remain relatively steady. Performance for rule 2 (threats) and rule 7 (non-Croatian language) may even be improving, although these rules have smaller amounts of data. Some of the main categories that relate to offensive language therefore seem to remain relatively consistent, while other categories such as advertising, spam and distribution of obscene content may be changing more. This may be because topics and vocabulary change over time; because authors change their language to avoid detection; because moderators change their criteria and behaviour; or a combination of these factors. What seems clear is that change over time is a significant issue: the ability to re-train classifiers on new data and up-to-date moderation labels will be important in practice.

5. Discussion and Conclusions

In this section we discuss the possible reasons for the overall levels of performance observed, and draw conclusions about what steps can be taken to improve it.

Year	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
2016	52.37	65.00	75.85	46.07	46.77	93.51	63.96	78.62
2017	49.36	76.92	70.27	51.68	46.99	85.71	71.21	73.34
2018	50.67	83.54	71.74	42.74	37.74	90.93	38.20	68.73
2019	64.23	66.67	72.18	100.00	54.36	88.32	35.85	72.17

Table 15: Blocking rule classification performance over the yearwise testset using mBERT, measured as percentage accuracy, 24sata.

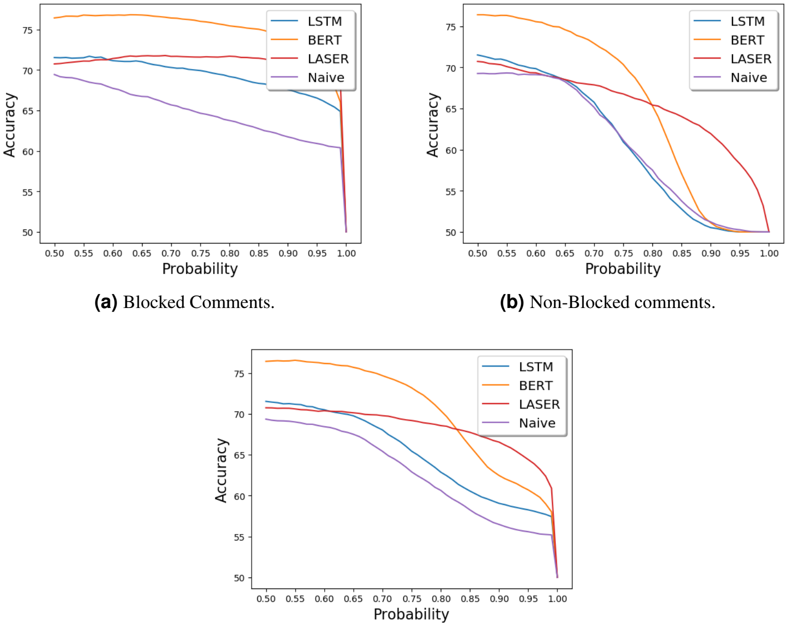
5.1. Analysis of Classifier Outputs

Figure 3 shows the confidence of the different classifier models: the plots are generated by changing the decision threshold of each classifier, increasing from the default 0.5 up to 1.0, and calculating the classification accuracy on the standard 24sata test set of Section 4.2. This is shown for blocked comments in Figure 3a, for non-blocked comments in Figure 3b, and the overall average in Figure 3c. The BERT and LASER models show overall higher confidence: increasing the threshold at which the decision is made has less effect on the accuracy of their output. The NB and to a lesser extent LSTM models' performance drops off more quickly, showing that their outputs give lower confidences for many correct classification decisions. Interestingly, classifier confidences seem significantly higher for blocked comments: the dropoff in performance is much less than that for non-blocked comments as the threshold increases. Although its performance was generally lower, the LASER model may provide some advantages here: its confidence curve is flatter with less dropoff for non-blocked comments.

This general tendency suggests that non-blocked comments are harder to classify in many cases. This may be due to variability or lack of reliability in moderation, with many comments that should be blocked labelled as non-blocked. Classifiers would therefore be learning decision boundaries that fit these examples where possible, but having to leave them close to the boundary given their similarity to other blocked comments.

Manual inspection of classifier errors was carried out over a set of approximately 350 comments on which the best (mBERT) classifier output disagreed with the moderator's decisions. These comments were passed back to 24sata's moderators, who were asked to moderate them again and produce a new set of labels. Of 101 comments which were originally not blocked, the majority (82) were still not blocked, but with a significant proportion (19) now marked as blocked. The problem of moderators missing comments which should be blocked is therefore a real one, as suspected. However, a bigger effect may be the variability of moderation decisions. Of 244 comments which were originally blocked (but given a non-blocked label by our classifier), approximately half (124) were still judged to be blocked, but half (120) were now marked non-blocked. Of the 124 which remained blocked, over half (81) were given a different rule as justification for blocking.

Examination of the errors also helps shed some light on the phenomena which cause difficulties for automatic classification. Some examples show classic language processing problems: non-standard spelling and vocabulary, and complex references and indirect statements can all be hard for classifiers to recognise without extremely large training sets. Two particular phenomena emerge as covering a large proportion of examples, however. One is that reader comments occur in the context of the article and the preceding comments, and many references need that context to be understood (see example (1), in which the phrase "that symbol" refers to an important concept from the previous discussion, probably the swastika. Treating comments as independent texts (as we do here) misses this – without the reference, it is hard to understand the comment as problematic. The second is that many comments use culture- and country-specific references which must also be resolved before the stance of the comment is clear. Example (2) appears on the face of it as a political trolling attempt; but if one knows that the HDZ and SDP are not only opposing political parties, but the only two large parties in Croatia, it can be understood as even-handed. In example (3), one must know that Pavelić headed a fascist government, and that



(a) Blocked Comments.

(b) Non-Blocked comments.

(c) Both comments.

Figure 3: Confidence of the Classifier.

Tuđman founded the currently governing, right-of-centre HDZ, in order to see its provocative nature.

(1) U čemu je problem? Dotični je pod tim simbolom živio i djelovao.
What's the problem? The person in question lived and worked under that symbol.
Moderator decision: blocked, rule 8

(2) HDZ je proslost a i Sdp !
HDZ is the past, and so is the SDP!
Moderator decision: not blocked

(3) Naime, preko natpisa "Franjo Tuđman, prvi hrvatski predsjednik"... Profesor Milan Kangrga je u emisiji NU2 rekao da je prvi hr pred bio Ante Pavelić.
Namely, via the inscription "Franjo Tuđman, the first Croatian president" ... Professor Milan Kangrga said on the NU2 show that the first Croatian president was Ante Pavelić.
Moderator decision: blocked, rule 8/rule 5 (moderators disagree)

5.2. Conclusions and Further Work

The high levels of variability in moderation decisions, and in the justifications given for them according to the moderation policy, indicate that an iterative approach may be of benefit in this task. Working with moderators to jointly define a more reliable policy, based partly on observation and use of high-confidence classifier outputs as in the error analysis above, would allow us to work towards less noisy data together with more reliable and useful classifiers. This could be framed within a general active learning approach, and we hope to explore this in future work. However, working within a real-world setting constrains the time and resources that can be dedicated to such work; great care must be taken to find an approach which does not further burden moderators and news publishers.

Second, the use of moderation flags as training labels, as pursued here and in other related work (Pavlopoulos et al., 2017a), may not be the most practical way to proceed in order to produce an accurate classification tool. A more effective and reliable way may be to use other, better-understood and curated datasets which represent the categories of language and author behaviour which should be blocked. By training classifiers on these cleaner datasets, a more reliable set of classifier outputs may be obtained which can feed into an active learning approach as outlined above. However, as Section 3 explains, such datasets are simply not available in the languages of interest here (Croatian and Estonian), or in many other language other than the majority well-resourced languages such as English, German and Spanish. One helpful step might be to pre-train word embeddings and/or models on data in the target language, even if annotated data is not available, to help smooth the noise from the training set; but note that the LASER and BERT models used here already benefit from large amounts of multi-lingual data, and in any case this is unlikely to go far towards solving the problem. Cross-lingual approaches (Ruder et al., 2017) would therefore be of great benefit if they can permit transfer learning from well-understood datasets in better-resourced languages to tasks in less-resourced languages.

However, while some work in hate speech and offensive language detection has been multilingual, studying datasets in more than one language, cross-lingual work is rare. A. Basile & Rubagotti (2018) use a *bleaching* approach (van der Goot et al., 2018) to conduct cross-lingual experiments between Italian and English in the EVALITA 2018 misogyny identification task, and Pamungkas & Patti (2019) propose a cross-lingual approach using a LSTM joint-learning model with multilingual MUSE embeddings. However, as far as we are aware, no work has yet tried to apply this to the problem of comment filtering, or focused on the languages needed here. As our error analysis shows, the task here poses significant challenges for cross-lingual techniques: many phenomena of interest are dependent on region- or culture-specific references and understanding of the related context, as in the need to understand country-specific relations between political parties and individuals discussed in the previous section. Current cross-lingual techniques depend on parallel corpus training, or on mapping of embedding spaces based on known synonymous anchor points (e.g. digits); these are unlikely to capture such phenomena well. Our next steps will therefore be to adapt techniques for cross-lingual learning to try to better map the entities, events and similar references found in news text between languages.

6. Acknowledgements

This research is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The work of AP was funded also by the European Union's Rights, Equality and Citizenship Programme (2014-2020) project IMSyPP (Innovative Monitoring Systems and Prevention Policies of Online Hate Speech, grant no. 875263). We acknowledge also the funding by the Slovenian Research Agency (ARRS) core research programme Knowledge Technologies (P2- 0103). The results of this publication reflect only the authors' views and the Commission is not responsible for any use that may be made of the information it contains.

References

Aiyar, S., & Shetty, N. P. (2018). N-gram assisted Youtube spam comment detection. *Procedia Computer Science*, *132*, 174 - 182. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1877050918309153` (International Conference on Computational Intelligence and Data Science) doi: https://doi.org/10.1016/j.procs.2018.05.181

Alberto, T., Lochter, J., & Almeida, T. (2015, December). TubeSpam: Comment spam filtering on YouTube. In *Proceedings of the 14th ieee international conference on machine learning and applications (icmla'15)* (p. 1-6).

Álvarez-Carmona, M. Á., Guzmán-Falcón, E., Montes-y Gómez, M., Escalante, H. J., Villasenor-Pineda, L., Reyes-Meza, V., & Rico-Sulayes, A. (2018). Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets. In *Notebook papers of 3rd sepln workshop on evaluation of human language technologies for iberian languages (ibereval), seville, spain* (Vol. 6).

Anzovino, M., Fersini, E., & Rosso, P. (2018). Automatic identification and classification of misogynistic language on Twitter. In M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, & F. Meziane (Eds.), *Natural language processing and information systems (NLDB)* (Vol. 10859, p. 57-64). Springer.

Artetxe, M., & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, *7*, 597–610.

Badawy, A., Ferrara, E., & Lerman, K. (2018, Aug). Analyzing the digital traces of political manipulation: The 2016 Russian interference Twitter campaign. In *Ieee/acm international conference on advances in social networks analysis and mining (asonam)* (p. 258-265). doi: 10.1109/ASONAM.2018.8508646

Barker, E., Paramita, M. L., Aker, A., Kurtic, E., Hepple, M., & Gaizauskas, R. (2016, September). The SENSEI annotated corpus: Human summaries of reader comment conversations in on-line news. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue* (pp. 42–52). Los Angeles: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W16-3605` doi: 10.18653/v1/W16-3605

Basile, A., & Rubagotti, C. (2018). Crotonemilano for ami at evalita2018. a performant, cross-lingual misogyny detection system. In *Evalita@ clic-it.*

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., … Sanguinetti, M. (2019, June). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proc. semeval* (pp. 54–63). Retrieved from `https://www.aclweb.org/anthology/S19-2007` doi: 10.18653/v1/S19-2007

Caruana, G., & Li, M. (2012, March). A survey of emerging approaches to spam filtering. *ACM Computing Surveys*, *44*(2), 9:1–9:27. Retrieved from `http://doi.acm.org/10.1145/2089125.2089129` doi: 10.1145/2089125.2089129

Chen, C., Zhang, J., Chen, X., Xiang, Y., & Zhou, W. (2015, June). 6 million spam tweets: A large ground truth for timely Twitter spam detection. In *2015 ieee international conference on communications (icc)* (p. 7065-7070). doi: 10.1109/ICC.2015.7249453

de Gibert, O., Perez, N., García-Pablos, A., & Cuadros, M. (2018, October). Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd workshop on abusive language online (ALW2)* (pp. 11–20). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W18-5102` doi: 10.18653/v1/W18-5102

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Gautam, K., Benevenuto, F., … Gummadi, K. (2012, April). Understanding and Combating Link Farming in the Twitter Social Network. In *Proceedings of the 21st International World Wide Web Conference (WWW'12).* Lyon, France.

Hochreiter, S., & Schmidhuber, J. (2015). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Jurafsky, D., & Martin, J. (2009). *Speech and language processing* (2nd ed.). Pearson Prentice Hall.

Kantchelian, A., Ma, J., Huang, L., Afroz, S., Joseph, A. D., & Tygar, J. D. (2012, October). Robust detection of comment spam using entropy rate. In *Proceedings of the 5th acm workshop on artificial intelligence and security* (p. 59-70).

Kim, D., Graham, T., Wan, Z., & Rizoiu, M. (2019). Tracking the digital traces of Russian trolls: Distinguishing the roles and strategy of trolls on Twitter. *CoRR*, *abs/1901.05228*. Retrieved from `http://arxiv.org/abs/1901.05228`

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd international conference on learning representations (iclr).*

Kolhatkar, V., Wu, H., Cavasso, L., Francis, E., Shukla, K., & Taboada, M. (2019, Nov 02). The sfu opinion and comments corpus: A corpus for the analysis of online news comments. *Corpus Pragmatics*. Retrieved from `https://doi.org/10.1007/s41701-019-00065-w` doi: 10.1007/s41701-019-00065-w

Linvill, D. L., & Warren, P. L. (2018). *Troll factories: The internet research agency and state-sponsored agenda building.* Retrieved from `http://pwarren.people.clemson.edu/Linvill_Warren_TrollFactory.pdf`

Liu, P., Guberman, J., Hemphill, L., & Culotta, A. (2018). Forecasting the presence and intensity of hostility on instagram using linguistic and social features. In *Twelfth international aaai conference on web and social media.*

Ljubešić, N., Erjavec, T., & Fišer, D. (2018, October). Datasets of Slovene and Croatian moderated news comments. In *Proc. 2nd workshop on abusive language online* (pp. 124–131). Retrieved from `https://www.aclweb.org/anthology/W18-5116` doi: 10.18653/v1/W18-5116

Ljubešić, N., Fišer, D., & Erjavec, T. (2019). The FRENK datasets of socially unacceptable discourse in Slovene and English. *CoRR*, *abs/1906.02045*. Retrieved from `http://arxiv.org/abs/1906.02045`

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019, 08). Hate speech detection: Challenges and solutions. *PLOS ONE*, *14*(8), 1-16. Retrieved from `https://doi.org/10.1371/journal.pone.0221152` doi: 10.1371/journal.pone.0221152

Mihaylov, T., & Nakov, P. (2016, August). Hunting for troll comments in news community forums. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 399–405). Berlin, Germany: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/P16-2065` doi: 10.18653/v1/P16-2065

Mojica, L. G. (2017). A trolling hierarchy in social media and A conditional random field for trolling detection. *CoRR*, *abs/1704.02385*. Retrieved from `http://arxiv.org/abs/1704.02385`

Mojica de la Vega, L. G., & Ng, V. (2018, May). Modeling trolling in social media conversations. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC-2018).* Miyazaki, Japan: European Languages Resources Association (ELRA). Retrieved from `https://www.aclweb.org/anthology/L18-1585`

Napoles, C., Tetreault, J., Rosata, E., Provenzale, B., & Pappu, A. (2017, April). Finding good conversations online: The yahoo news annotated comments corpus. In *Proceedings of the 11th linguistic annotation workshop* (pp. 13–23). Valencia, Spain: Association for Computational Linguistics.

Narayanan, A. (2018). *Tweets dataset for detection of cyber-trolls.* Retrieved from `https://www.kaggle.com/dataturks/dataset-for-detection-of-cybertrolls`

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., & Yeung, D.-Y. (2019). Multilingual and multi-aspect hate speech analysis. *arXiv preprint arXiv:1908.11049*.

Pamungkas, E. W., & Patti, V. (2019). Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop* (pp. 363–370).

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017a, September). Deeper attention to abusive user content moderation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1125–1135). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/D17-1117` doi: 10.18653/v1/D17-1117

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017b, August). Deep learning for user comment moderation. In *Proceedings of the first workshop on abusive language online* (pp. 25–35). Vancouver, BC, Canada: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W17-3004` doi: 10.18653/v1/W17-3004

Ruder, S., Vulić, I., & Søgaard, A. (2017). A survey of cross-lingual word embedding models. *CoRR*, *abs/1706.04902*. Retrieved from `http://arxiv.org/abs/1706.04902`

Schmidt, A., & Wiegand, M. (2017, April). A survey on hate speech detection using natural language processing. In *Proceedings of the 5th international workshop on natural language processing for social media* (pp. 1–10). Retrieved from `https://www.aclweb.org/anthology/W17-1101` doi: 10.18653/v1/W17-1101

van der Goot, R., Ljubešić, N., Matroos, I., Nissim, M., & Plank, B. (2018). Bleaching text: Abstract features for cross-lingual gender prediction. *arXiv preprint arXiv:1805.03122*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (p. 5998-6008).

Waseem, Z., & Hovy, D. (2016, 01). Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the naacl student research workshop* (p. 88-93).

Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018, September). Overview of the GermEval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval 2018, 14th conference on natural language processing (KONVENS).* Vienna, Austria.

Wu, T., Wen, S., Xiang, Y., & Zhou, W. (2018). Twitter spam detection: Survey of new approaches and comparative study. *Computers and Security*, *76*, 265-284. Retrieved from `http://www.sciencedirect.com/science/article/pii/S016740481730250X` doi: https://doi.org/10.1016/j.cose.2017.11.013

Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web* (pp. 1391–1399).

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). *CoRR*, *abs/1903.08983*. Retrieved from `http://arxiv.org/abs/1903.08983`

Zhang, J., Chang, J., Danescu-Niculescu-Mizil, C., Dixon, L., Hua, Y., Taraborelli, D., & Thain, N. (2018, July). Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1350–1361). Melbourne, Australia: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/P18-1125` doi: 10.18653/v1/P18-1125

A. Yearwise Data Distribution

This section gives the full details of the dataset distributions over time, in terms of overall numbers of articles, comments and moderator's blocking behaviour for all three newspapers (Section 4.2.1), and the frequency of application of individual blocking rules for 24sata and VL (Section 4.4).

A.1. Summary data, commenting and blocking rates

Year	# Articles	# Comments	# Blocked	Comment rate	Blocking rate
2007	6054	38005	3	6.3	7.9×10^{-5}
2008	26523	185578	12	7.0	6.5×10^{-5}
2009	38024	326609	31	8.6	9.5×10^{-5}
2010	38777	459227	2	11.8	4.4×10^{-6}
2011	38330	1140555	111	29.8	9.7×10^{-5}
2012	43978	1870449	251	42.5	1.3×10^{-4}
2013	46457	2490285	130	53.6	5.2×10^{-5}
2014	46429	2656841	171	57.2	6.4×10^{-5}
2015	44919	3054087	724	68.0	2.4×10^{-4}
2016	47595	3194761	98487	67.1	3.1×10^{-2}
2017	45891	2795824	134080	60.9	4.8×10^{-2}
2018	48777	2519279	156083	51.7	6.2×10^{-2}
2019	17953	816692	63972	45.5	7.8×10^{-2}
Total	489707	21548192	454057		

Table 16: Yearwise data distribution, 24sata; comment rate $= N_{\text{comments}}/N_{\text{articles}}$, blocking rate $= N_{\text{blocked}}/N_{\text{comments}}$.

Year	# Articles	# Comments	# Blocked	Comment rate	Blocking rate
2009	7724	162017	4	20.98	2.47×10^{-5}
2010	31423	764134	175	24.32	2.29×10^{-4}
2011	32521	1245946	91	38.31	7.30×10^{-5}
2012	35693	1022186	29	28.64	2.84×10^{-5}
2013	41408	1101234	16747	26.59	1.52×10^{-2}
2014	43251	835152	48099	19.31	5.76×10^{-2}
2015	43469	1237714	48930	28.47	3.95×10^{-2}
2016	40485	1009070	60390	24.92	5.98×10^{-2}
2017	38136	840677	87476	22.04	1.04×10^{-1}
2018	42092	1073953	130054	25.51	1.21×10^{-1}
2019	16453	354551	55295	21.55	1.56×10^{-1}
Total	372655	9646634	447290		

Table 17: Yearwise data distribution, Večernji List; comment rate $= N_{\text{comments}}/N_{\text{articles}}$, blocking rate $= N_{\text{blocked}}/N_{\text{comments}}$.

Year	# Articles	# Comments	# Blocked	Comment rate	Blocking rate
2009	109352	2898438	130040	26.51	4.49×10^{-2}
2010	105173	2377591	107735	22.61	4.53×10^{-2}
2011	127037	2729389	148302	21.49	5.43×10^{-2}
2012	127663	3372776	249880	26.42	7.41×10^{-2}
2013	114914	3289393	295608	28.63	$8.99\ \times10^{-2}$
2014	101936	3195502	336450	31.35	10.53×10^{-2}
2015	98198	3202592	391758	32.61	12.23×10^{-2}
2016	94353	2848624	355868	30.19	12.49×10^{-2}
2017	87098	2838075	265810	32.58	9.37×10^{-2}
2018	82887	3194597	343538	38.54	10.75×10^{-2}
2019	32691	1540382	188197	47.12	12.21×10^{-2}
Total	1081302	31487359	2813186		

Table 18: Yearwise data distribution, Ekspress; comment rate $= N_{\text{comments}}/N_{\text{articles}}$, blocking rate $= N_{\text{blocked}}/N_{\text{comments}}$.

A.2. Blocking rule distribution

	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
2007						1		2
2008	12							
2009	29		1					1
2010	2							
2011	107							4
2012	144				2	9	13	83
2013	112				5		1	12
2014	108	1	1		45	2		14
2015	659	2	7		18	1		37
2016	23551	111	3152	183	2479	7400	227	61384
2017	50178	185	5310	153	4631	5752	137	67734
2018	65775	254	8099	125	8483	3453	780	69114
2019	31592	26	2734	37	3658	1270	498	24157

Table 19: Yearwise blocking rule data distribution, 24sata.

	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8	Rule9
2009		4							
2010	91		1		1				82
2011	44		1		4				42
2012	8		4	2	4				11
2013	1748	52	6575	16	7192	15	618	275	256
2014	4913	83	20911	19	16462	114	813	142	4642
2015	5438	82	16729	24	21858	109	187	4	4499
2016	4859	118	14007	10	38076	147	889	2	2282
2017	28888	169	15251	30	35957	195	608		6378
2018	33660	8076	17311	4	37572	45	256	8	33122
2019	4860	8477	5748	72	4712	11	199	4	31212

Table 20: Yearwise blocking rule data distribution, Večernji List.

B. Word Lists

This section gives the full top 100 words lists for blocked and non-blocked comments as inferred by the Naïve Bayes classifier trained on the binary classification task (Section 4.2.2).

Word	Ratio	Word	Ratio	Word	Ratio
20544	2.16	ponio	1.50	jebo	1.40
22000	2.14	ovom	1.48	odnio	1.40
17000	2.14	ovog	1.48	ženu	1.40
pridružio	2.08	želiš	1.47	sada	1.40
mreži	2.08	internetu	1.47	dobivanje	1.40
www	2.03	radno	1.46	nepunim	1.39
com	2.02	jebem	1.46	redoviti	1.39
mjesečno	1.94	promijenilo	1.45	pogledam	1.39
google	1.94	slijedite	1.45	radeci	1.39
mjesecu	1.85	dnevno	1.45	sponzoru	1.39
kuće	1.81	paycheck	1.44	šokiran	1.38
dolara	1.80	eura	1.44	redovne	1.38
mjeseca	1.79	odlučio	1.44	počeo	1.38
prvom	1.78	dnevne	1.44	stanicom	1.38
poslu	1.77	nabijem	1.43	odabirete	1.38
zaradio	1.76	litte	1.43	primio	1.38
rad	1.74	24857	1.43	vremenom	1.37
radeći	1.70	čula	1.43	zarađivati	1.37
promijenjen	1.69	web	1.42	želite	1.36
plaća	1.69	top	1.42	blogu	1.36
dobrodošli	1.69	započela	1.42	prije	1.36
7645	1.67	premise	1.42	dodatni	1.36
9264	1.67	rasponu	1.42	86	1.36
27936	1.67	prošlog	1.42	prethodni	1.36
tjedno	1.57	počinjem	1.41	zaradite	1.35
online	1.57	četiri	1.41	rate	1.35
pronaći	1.55	jednostavan	1.41	39	1.35
mom	1.54	29584	1.41	stranicu	1.35
posla	1.53	22738	1.41	posjetite	1.35
zaraditi	1.53	sam	1.41	majmune	1.35
noć	1.52	debil	1.40	mijenjam	1.34
skraćeno	1.52	računalo	1.40	govno	1.34
satu	1.51	jo	1.40	nepuno	1.34
				mjesec	1.34

Table 21: Top 100 word features for blocked comments, in order of class probability ratio

Word	Ratio	Word	Ratio	Word	Ratio
sritno	1.26	vrtić	1.18	gripa	1.16
nii	1.25	noja	1.18	kapetan	1.16
sretno	1.24	liniju	1.18	ličnost	1.16
strašno	1.24	tekma	1.17	težak	1.16
inter	1.23	ponovilo	1.17	niš	1.16
derbi	1.21	šanse	1.17	sudar	1.16
napišite	1.21	osijek	1.17	petak	1.16
naklon	1.21	strah	1.17	bok	1.16
malena	1.20	ajmoo	1.17	vrhova	1.16
var	1.20	vozac	1.17	cirkusanti	1.16
štima	1.20	miša	1.17	šubi	1.16
zavisi	1.20	nima	1.17	terorizam	1.16
humbla	1.20	glumac	1.17	probaju	1.16
điri	1.20	kiša	1.17	jela	1.16
prekrasna	1.20	miru	1.17	sjeveru	1.16
svašta	1.20	išlo	1.17	cudimo	1.16
pocelo	1.20	vakula	1.17	potpisujem	1.16
počivaj	1.19	svizac	1.17	nadje	1.16
gledanost	1.19	dvojno	1.17	cares	1.16
drž	1.19	pila	1.17	žiri	1.16
oja	1.19	zasluženo	1.17	hrabro	1.16
horor	1.19	ligama	1.17	kip	1.16
predivno	1.19	najte	1.17	blagi	1.16
obožavam	1.19	tragedija	1.17	dizel	1.16
mokra	1.19	baš	1.17	tuzno	1.16
odlično	1.18	teško	1.17	nasmijao	1.16
sumljam	1.18	skupit	1.17	informaciji	1.16
pocivao	1.18	troše	1.17	srećom	1.16
pravna	1.18	anđeli	1.17	trolaš	1.16
sućut	1.18	svastiku	1.17	prolazak	1.16
bisera	1.18	hep	1.17	lepi	1.16
ludost	1.18	najljepša	1.17	pretjerao	1.16
filmova	1.18	izvoli	1.16	čekala	1.16
				snijeg	1.16

Table 22: Top 100 word features for non-blocked comments, in order of class probability ratio

Author Index

Christine Carr
Department of Linguistics
University of North Texas
alexis.palmer@unt.edu

Ralf Krestel
Hasso Plattner Institute, University of Potsdam, University of Passau
ralf.krestel@hpi.de

Alexis Palmer
Department of Linguistics
University of North Texas
alexis.palmer@unt.edu

Andraž Pelicon
Department of Knowledge Technologies
Institut Jožef Stefan
Ljubljana, Slovenia
andraz.pelicon@ijs.si

Senja Pollak
Department of Knowledge Technologies
Institut Jožef Stefan
Ljubljana, Slovenia
senja.pollak@ijs.si

Marko Pranjić
Trikoder d.o.o.
Zagreb, Croatia
marko.pranjic@styria.ai

Matthew Purver
Cognitive Science Research Group,
School of Electronic Engineering and Computer Science
Queen Mary University of London, UK
AND:
Department of Knowledge Technologies
Institut Jožef Stefan
Ljubljana, Slovenia
m.purver@qmul.ac.uk

Julian Risch
Hasso Plattner Institute, University of Potsdam
julian.risch@hpi.de

Melissa Robinson
Department of Linguistics
University of North Texas
alexis.palmer@unt.edu

Robin Ruff
University of Passau, Karlsruhe Institute of Technology
upnub@student.kit.edu

Jordan Sanders
Department of Computer Science and Engineering
University of North Texas
JordanSanders3@my.unt.edu

Ravi Shekhar
Cognitive Science Research Group
School of Electronic Engineering and Computer Science
Queen Mary University of London, UK
r.shekhar@qmul.ac.uk