Benchmarking Offensive and Abusive Language in Dutch Tweets

Tommaso Caselli and **Hylke van der Veen**
CLCG, University of Groningen
t.caselli@rug.nl | hylkevdveen@gmail.com

Abstract

We present an extensive evaluation of different fine-tuned models to detect instances of offensive and abusive language in Dutch across three benchmarks: a standard held-out test, a task-agnostic functional benchmark, and a dynamic test set. We also investigate the use of data cartography to identify high quality training data. Our results show a relatively good quality of the manually annotated data used to train the models while highlighting some critical weakness. We have also found a good portability of trained models along the same language phenomena. As for the data cartography, we have found a positive impact only on the functional benchmark and when selecting data per annotated dimension rather than using the entire training material.

1 Introduction

Being able to correctly detect instances of offensive and abusive language plays a pivotal role in creating safer and more inclusive environments, especially on Social Media platforms. Since current methods for these phenomena are based on supervised techniques, a pending issue is represented by the quality of the data used to train the corresponding systems. Standard evaluation methods based on held-out test sets only provide a partial picture of the actual robustness of fine-tuned models while being silent about potential annotators' bias, topic and author biases (Wiegand et al., 2019). Recent work has show that held-out tests may result in overly optimistic performance estimates which do not translate into real-world performance (Gorman and Bedrick, 2019; Søgaard et al., 2021). To get a realistic performance estimate, models should be evaluated on out-of-corpus data, i.e. a different data distribution but within the same language variety (Ramponi and Plank, 2020), or even on a held-out test set from a different but related domain. Out-of-corpus evaluation requires the development of multiple datasets which can be expensive, time consuming, and, in the case of less- or poor-resources languages, unfeasible.

A complementary solution is the use of functional tests, i.e., sets of systematically generated test cases aiming at evaluating in a task-agnostic methodology trained models (Ribeiro et al., 2020; Lent et al., 2021; Sai et al., 2021; Röttger et al., 2021; Manerba and Tonelli, 2021). Functional testing enables more targeted insights and diagnostics on multiple levels. For instance, the systematic categorisation as hateful of messages containing a protected identity term (e.g., "gay", "trans", among others) of a system trained to detect hate speech against LGBTQIA+ people is an indicator of the weakness of the model(s) as well as of biases in the training data.

Although limited in terms of number of datasets and annotated phenomena, Dutch covers a peculiar position in the language resource panorama: it has a comprehensively annotated corpus for offensive and abusive language whose standard held-out test set does not present any overlap with the training set; it includes a dynamic benchmark for offensive language, OP-NL (Theodoridis and Caselli, 2022); and it presents a functional benchmark, HATECHEK-NL, that extends MULTILINGUAL HATECHEKCK (Röttger et al., 2022). This puts us is an optimal position to conduct an extensive benchmarking of different models for offensive and abusive language in Dutch and reflect on the potential shortcomings of the Dutch Abusive Language Corpus v2.0 (DALC-V2.0) (Ruitenbeek et al., 2022). In addition to this, we apply data cartography (Swayamdipta et al., 2020) to carve out different subsets of training materials to investigate whether this method is valid on DALC-V2.0 to identify robust and good quality training data.

Our contributions Our major contributions are the followings: (i) we present and discuss our ex-

The 7th Workshop on Online Abuse and Harms (WOAH), pages 69–84
July 13, 2023 ©2023 Association for Computational Linguistics

tensions of HATECHEK-NL (Section 2); (ii) we apply data cartography (Swayamdipta et al., 2020) to DALC-v2.0 to investigate whether we can identify robust subsets of training data (Section 3); (iii) we conduct an extensive evaluation of different systems based on a monolingual pre-trained language model, namely BERTje (de Vries et al., 2019), against multiple test sets (Section 4).[1]

2 Data

In this section, we present the data we use to fine-tune and evaluate the models based on BERTje (de Vries et al., 2019).

DALC-v2.0 DALC-v2.0 contains 11,292 messages from Twitter in Dutch, covering a time period between November 2015 and August 2020. Messages have been annotated using a multi-layer annotation scheme compliant with Waseem et al. (2017) for two dimensions: offensive and abusive language. Offensive language in DALC-v2.0 is the same as in Zampieri et al. (2019), i.e., messages "containing any form of non-acceptable language (profanity) or a targeted offence, which can be veiled or direct". Abusive language corresponds to "impolite, harsh, or hurtful language (that may contain profanities or vulgar language) that result in a debasement, harassment, threat, or aggression of an individual or a (social) group, but not necessarily of an entity, an institution, an organisations, or a concept." (Caselli et al., 2021, 56–57). Each dimension is further annotated along two layers: explicitness and target. The explicitness layer is used to annotate whether a message is belonging to the positive category or not. In the former case, the values explicit (EXP) and implicit (IMP) are used to distinguish the way the positive category is realised. The target layer is used to annotate towards who or what the offence, or abuse, is directed to. Target layers inherit values from Zampieri et al. (2019), namely individual (IND), group (GRP), other (OTH).

Here we focus only on the explicitness layer, considering each dimension separately and jointly. In particular, when addressing each dimension separately, we frame the task as a binary classification by collapsing the explicit and implicit labels either into OFF and ABU for the offensive and abusive dimension, respectively. When working on both dimensions jointly, we face a multi-class classification where systems must distinguish between two positive classes (OFF and ABU) and one negative (NOT). Table 1 illustrates the distribution of the data for the dimensions in analysis across the Train/Dev and standard held-out test splits.

Annotated Dimension	Label	Train	Dev	Test	Total
Offensive	OFF	2,477	439	867	3,783
	NOT	4,340	766	2,403	7,509
Abusive	ABU	1,391	243	463	2,097
	NOT	5,426	962	2,807	9,195
Offensive & Abusive	OFF	1,086	196	404	1,686
	ABU	1,391	243	463	2,097
	NOT	4,304	766	2,403	7,473

Table 1: DALC-v2.0 : Distribution of labels (binary and multi-class settings) in Train, Dev, and official held-out Test splits for each annotated dimension independently and jointly.

Labels are skewed towards the negative class as in previous work (Basile et al., 2019; Davidson et al., 2017; Zampieri et al., 2019, 2020). When considering each dimension separately, the offensive dimension is larger than the abusive one (*approx* 33% of the total *vs.* ≈ 19%, respectively). In the joint setting, the OFF messages drop to ≈ 15%. This reflects the definitions of offensive and abusive language and how the two phenomena interact: abusive language is more specific and subject to a stricter set of criteria for its identification (e.g., a target must always be present), resulting in a "specialized instance" of offensive language (Poletto et al., 2020). In other words, while every abusive message is also offensive, the contrary does not hold. In their analysis of the corpus, the authors do not report evidence of any specific topic bias and they state that train and test splits have no overlap (Caselli et al., 2021; Ruitenbeek et al., 2022).

HATECHEK-NL HATECHEK-NL extends MULTILINGUAL HATECHEKCK (MHC) (Röttger et al., 2022). MHC defines hate speech as "abuse that is targeted at a protected group or at its members for being a part of that group." (Röttger et al., 2022, 155). This definition is more specific than the language phenomena in DALC-v2.0, although it is compatible. MHC has 27 common functionalities for 10 languages, including Dutch, 18 specific for *expressions of hate* and nine non-hateful to *contrast the hateful cases*. Each test is realised by a short text uniquely identifying a gold label (e.g.,

[1] All code, data, and trained models are available via https://github.com/tommasoc80/DALC

hateful *vs.* non-hateful). To massively generate tests, MHC makes use of templates (Ribeiro et al., 2020). We have extended the functionalities in MHC with two extra tests to include the use of reclaimed slurs and profanities in a non hateful way (**F8**, **F9**). These two functional tests are present in the original English HATECHECK (Röttger et al., 2021) but they were excluded from MHC to maintain a more homogeneous distribution of functional tests across all languages. Röttger et al. (2022) observe that these functionalities have no direct equivalents in most of the languages in MHC, but this is not the case for Dutch. For the functionality **F8** (non-hateful homonyms of slurs), we have identified four slurs that are each aimed at one of the target identities and have a non-hateful homonym. For instance, the term "f*****r" is used to refer to gay men or as a verb meaning flickering of a light, to fall or to drop something. Reclaimed slurs (**F9**) have been partially translated from English, excluding terms such as "n****r" and "b***h" for which we have not found evidence of their use in Dutch nor have we identified corresponding terms.

HATECHEK-NL contains 3,835 functional tests across the 29 functionalities. A total of 2,640 (68.83%) tests are hateful and 1,195 (31.16%) are non-hateful, a distribution in line with the original HATECHECK. An overview of all the functionalities in HATECHEK-NL is in Table A.1 in Appendix A. On the basis of the annotated dimensions in DALC-V2.0, we expect that models trained on offensive language may overgeneralise the identification of hateful messages, also for challenging non-hateful cases (e.g., **F8**, **F9**). On the other hand, we expect models trained on abusive language (both in isolation and jointly) to perform better, although the emphasis on "protected group and its members" in HATECHEK-NL may present an extra challenge since no specific protected group is part of DALC-V2.0.

OP-NL Offend the Politicians Benchmark (OP-NL) is a dynamic test set composed by 1,500 tweets collected in March 2021 containing at least one mention of a Dutch politician from the *Tweede Kamer* (i.e., the Dutch House of Representatives). The messages have been annotated for offensive language using the same definition of DALC-V2.0, making OP-NL perfectly compatible and suitable as a dynamic benchmark. The labels in OP-NL are distributed as follows: 961 messages (64%) are not offensive (NOT) and 539 (36%) are offensive (OFF). The ratio between non-offensive and offensive messages is 1.78 : 1, very close to the label distribution in DALC-V2.0. In this case, we expect offensive language models (in isolation or jointly with abusive language) to obtain good performances, i.e., in-line with those on DALC-V2.0 for offensive language. On the contrary, models trained for abusive language are expected to struggle, mainly on the recall for the positive class.

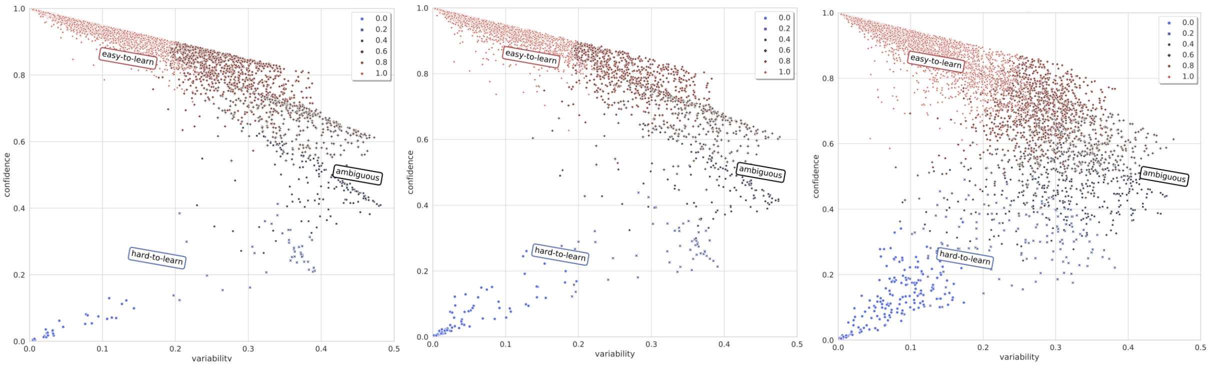
3 Experiment settings

We have designed three sets of experiments for each annotated dimension to fine-tune a monolingual pre-trained language model for Dutch, BERTje, with varying training splits. All fine-tuned models are evaluated both on the official DALC-V2.0 held-out test set, HATECHEK-NL, and OP-NL. All pre-processing steps and fine-tuning (hyper)parameters are detailed in Appendix B for replicability.

The first block of experiment has a standard setting: for each annotated dimension (in isolation or jointly) we fine-tuned BERTje using all available training data in DALC-V2.0. We will refer to these models as standard (**std**).

For the second block, we use data cartography (Swayamdipta et al., 2020). The cartography approach uses a model's confidence in the true class and the variability of this confidence across multiple training epochs (i.e., training dynamics) to identify a subset of training instances that qualify as more reliable and informative. In this way, it is possible to train a model using less data and still achieve state-of-the-art results, if not better. When plotting statistics from the training dynamics into a map, they result into a spectrum of data points: some *easy* (high-confidence, low variability), some *hard* (low-confidence, low variability), and some *ambiguous* (mid-range confidence, high variability). Previous work (Swayamdipta et al., 2020; Bhargava et al., 2021) has shown that, in classification tasks, the use of *ambiguous* data points at training time results in better models than those obtained when using the entire training split. Our goal is to test the validity of this method on DALC-V2.0, a smaller dataset than those where data cartography has been successfully applied.

To identify the ambiguous data points, we have used the training dynamics from the fine-tuned models from each classification task from DALC-V2.0. Given its skewed distribution and size, we



(a) Offensive data map

(b) Abusive data map

(c) Offensive and abusive data map

Figure 1: DALC-v2.0: data maps from training dynamics for each annotated dimension with BERTje.

Split	Dimension	Labels		Avg. Variability
amb-dim	Offensive	OFF	1,192	$0.255_{.089}$
		NOT	1,080	
	Abusive	ABU	1,136	$0.225_{.101}$
		NOT	1,136	
	Offensive & Abusive	OFF	894	$0.280_{.057}$
		ABU	714	
		NOT	664	
amb-class	Offensive	OFF	1,136	$0.123_{.120}$
		NOT	1,136	
	Abusive	ABU	1,136	$0.142_{.131}$
		NOT	1,136	
	Offensive & Abusive	OFF	757	$0.182_{.115}$
		ABU	757	
		NOT	758	

Table 2: Ambiguous train splits per annotated dimensions (**amb-dim**) or per class per dimension (**amb-class**). Numbers in subscript report standard deviations.

Split	Dimension	Labels		Avg. Variability
rand-1	Offensive	OFF	821	$0.114_{.116}$
		NOT	1,451	
	Abusive	ABU	458	$0.091_{.110}$
		NOT	1,814	
	Offensive & Abusive	OFF	363	$0.139_{.089}$
		ABU	458	
		NOT	1451	
rand-2	Offensive	OFF	814	$0.114_{.116}$
		NOT	1,458	
	Abusive	ABU	458	$0.094_{.112}$
		NOT	1,814	
	Offensive & Abusive	OFF	356	$0.147_{.097}$
		ABU	458	
		NOT	1,458	
rand-3	Offensive	OFF	855	$0.116_{.116}$
		NOT	1,417	
	Abusive	ABU	476	$0.095_{.112}$
		NOT	1,796	
	Offensive & Abusive	OFF	379	$0.143_{.087}$
		ABU	476	
		NOT	1,417	

Table 3: Random train splits (**rdm**) per annotated dimensions. Number in subscripts report standard deviations.

have investigated two methods to select the ambiguous data: the first (**amb-dim**) follows the approach in Swayamdipta et al. (2020) by retaining 1/3 of the original training data (i.e., 2,272 examples) corresponding to the top ambiguous cases per annotated dimension (separately and jointly). The second (**amb-class**) independently retains the top ambiguous examples for *each class*. In particular, we have carved three training splits of 2,272 examples where the distribution of instances per class is perfectly balanced (50-50 for binary settings, and 1/3 each for the multi-class setting). As the figures in Table 2 show, the class distribution is less skewed when compared to the original DALC-v2.0 training. For the abusive dimension, the distribution of the labels is perfectly balanced also when using the **amb-dim** method. The variability, across all data selection methods, is not particularly high. However, we observe a systematic difference between the values of the **amb-dim** and the **amb-class** data, with the latter being always lower of ≈ 0.1 points. Although in both cases the selected data instances qualifies as "ambiguous", the relatively low variability questions their efficacy as more robust training instances.

Figures 1a, 1b, and 1c illustrate the data maps of the training examples for the offensive and abusive dimension, separately and jointly. We can observe a consistent overlap between the easy and the ambiguous cases which questions the use of the ambiguous instances as effective training material from DALC-v2.0. At the same time, we observe that the hard examples are limited and well clustered for each dimension separately (Figures 1a and 1b), while this does not hold in the joint case (Figure 1c). In this case, the overlap between the hard and the ambiguous instances is larger, indicating, on one side, that the classification task is more challenging and, on the other side, that the distinction among the three classes is less clear than it seems.

The last set of training data has the same size of the ambiguous data (2,272 instances) but it is randomly extracted from the original training set (**rand**). It is a control to better asses the effectiveness of the data cartography on DALC-v2.0. Random splits have been sampled three times with different seeds and no substitution. Table 3 illustrates their distribution. In this case, the data are skewed towards the negative class and their variability is consistently lower than that of the ambiguous ones, suggesting that the corresponding fine-tuned models should obtain worst results.

4 Results

For the analysis of the results we first focus on DALC-v2.0, and subsequently on HateChek-nl and OP-NL. All fine-tuned models are compared against a baseline. For DALC-v2.0 and OP-NL, we use a dummy classifier that always assigns the most frequent class, i.e., NOT; for HateChek-nl, we use a random classifier (balanced for the hateful and non-hateful class distribution). The random classifier for HateChek-nl represents a more realistic baseline than a majority label classifier given the nature of the benchmark. Detailed results for each dataset are illustrated in Appendix C.

DALC-v2.0 Table 4 summarises the results on DALC-v2.0. All models largely outperform the baselines. When compared to previous work based on data cartography (Swayamdipta et al., 2020; Bhargava et al., 2021), we cannot find the same trends. Across all annotated dimensions and classification tasks (binary *vs.* multi-class), the use of the full training set (**std**) returns the best results, with a macro-F1 of 79.93 for offensive language, 72.33 for abusive language, and 58.90 for the two dimensions in conjunction. The identification of offensive and abusive language separately clearly returns better results than when the two dimensions are predicted jointly. This confirms the observations from the data maps (Figure 1c). In this latter case, the system mostly struggles to distinguish between the two positive classes. As it appears from the analysis of the predictions using a confusion matrix, for the abusive class the largest number of errors are messages classified as OFF (125 out of 463 instances), while for the offensive class most of the messages are wrongly classified either as ABU (137 out 404 instances) or as NOT (159 out 404 instances).

	DALC		
Train split	**Offensive**	**Abusive**	**Off. & Abu.**
baseline	42.35	46.19	28.24
std	**79.93**	**72.23**	**58.90**
amb-dim	68.85	66.31	43.74
amb-class	77.66	67.21	53.58
rdm	$77.64_{1.7}$	$70.70_{1.0}$	$57.26_{1.26}$

Table 4: Experiments results for each annotated dimension in DALC-v2.0 against the held-out test sets (per annotated dimension). Best scores per training split are marked in bold. Scores correspond to macro-F1. We report the average and standard deviations for the **rdm** splits.

The use of random subsets for training (**rdm**) is unexpectedly competitive when compared to the **std** split and both ambiguous subsets from the data maps. A better impact of selecting ambiguous data per class (**amb-class**) to generate balanced training sets is evident for all dimensions. A further unexpected behaviour is the better performances of low variability training sets (i.e., **amb-class** and **rdm**). While the results of the **amb-class** set may suggest a different way of selecting robust sub-samples using data maps, the **rdm** blocks question the validity of data maps with small datasets.

When narrowing down the analysis to the differences between the reduced training data, we identify a peculiar behaviour of the data map splits. In particular, **amb-dim** and **amb-class** tend to overgeneralise the positive classes, with higher recall values at the cost of precision. Given the distribution of the labels (see Table 2), it is difficult to explain this behaviour in terms of class imbalance. On the other hand, this effect appears to be directly related to the use of the data maps. The impression is that the selected training data for the positive classes are too "ambiguous" for the system resulting in overgeneralisations to the detriment (mainly) of the negative class. Support in this direction comes from the results of the **rdm** splits where precision and recall are more balanced.

HateChek-nl Table 5 reports the performances of the trained models on HateChek-nl.

	HATECHECK-NL			OP-NL		
Train Split	**Offensive**	**Abusive**	**Off. & Abusive**	**Offensive**	**Abusive**	**Off. & Abusive**
baseline	57.08	57.08	57.08	39.04	39.04	39.04
std	61.40	60.19	60.94	**73.56**	57.57	**71.85**
amb-dim	59.35	**62.72**	61.22	54.23	63.19	51.83
amb-class	**64.52**	62.42	**63.21**	69.91	**68.75**	66.41
rdm	$61.05_{19.56}$	$55.28_{20.55}$	$52.78_{26.96}$	$69.07_{0.83}$	$55.50_{4.28}$	$69.91_{2.51}$

Table 5: Results of the fine-tuned models against HATECHEK-NL and OP-NL. Best scores per model are in bold. Scores correspond to Accuracy for HATECHEK-NL and macro-F1 for OP-NL. We report the average and standard deviation for the **rdm** splits.

At evaluation time, for the joint model we have considered valid only the predictions for the ABU class, with the OFF labels as non-hateful messages.

In general, all fine-tune models outperform the baseline with the exceptions of the models fine-tuned on the **rdm** training data for abusive language and for offensive and abusive language jointly.

Models fine-tuned on offensive language obtain a better global accuracy. The sole deviation is represented by the model fine-tuned using the **amb-dim** data (59.35). This is mainly due to an overgeneralisation of the positive class in each functional test due to the broader and encompassing definition of offensive language. Being HATECHEK-NLunbalanced for the hateful labels, this gives the false impression of dealing with better models. To put things in perspective, consider that the average accuracy based on the majority label (i.e., all hateful) would be 68.83% - a score that no fine-tuned model can beat. Furthermore, these models fail the majority of the non-hateful functional tests, as we have predicted: in this cases, the accuracy ranges from 28.77% for **amb-class** to 52.57% for **rdm**, with only the model fine-tuned on **rdm** being above 50% (see also Table C.1). In particular, for the most challenging non-hateful tests, such as **F9** (reclaimed slurs), **F11** (not hateful use of profanities), **F21** (quotation of hate speech to counteract hate speech), **F23–24** (non hateful messages with individual or group targets), the accuracy is consistently below 50% across all training splits. At the same time, this is an indirect positive feedback on the quality of the annotation for offensive language in DALC-V2.0: the non-hateful tests may contain language and expressions that can be perceived as offensive, and thus are flagged by the models. This is particular evident with the results for **F11** where accuracy ranges between 15% and 33.67% since the presence of a profanity is flagged as offensive.

As for the use of abusive language as training, models have a more balanced behaviour between the hateful and the non-hateful cases. In particular, across all non-hateful tests, accuracy ranges from 36.29% for **amb-dim** to 65.72% for **rdm**, with one extra model, **std**, being above 50% (see Table C.2). For the challenging non-hateful tests, there is only one case where the performance is consistently below 50% across all training splits, namely **F16** (hate expressed via a question). For all the other non-hateful tests, the behaviour of the models is more varied with at least one or two models achieving results above 50%. To make a direct comparison with the offensive training splits, on **F9** and **F11** only two out four models are below 50% (**amb-dim**, and **amb-class**), while on **F21** and **F23–24**, three out of four are below 50% (**std**, **amb-dim**, and **amb-class**). In addition, the accuracy of these models is consistently higher when compared to their counterparts fine-tuned using offensive language. Again, this provides an indirect feedback on the quality of the annotated data and the compatibility of the definition of abusive language in DALC-V2.0 with that of hate speech in HATECHEK-NL. The results for **std** and **rdm** on **F9–F11** are particularly relevant. These functional tests are very useful to assess the generalisation functionalities of fine-tune models to distinguish between abusive/hateful content and the mere presence of slurs or swear words. Although half of the models achieve a score which is higher than 50%, there is still room for improvement: the best results for **F9** is only 66.70% (with **std**) and that for **F11** is 62.67 (with **rdm)**.

When focusing on the joint models, the picture that emerges is more complex than it seems at a first look. First, the joint models have a lower overall accuracy. Yet, these are the models that achieve the best results for all non-hateful tests, with the accuracy ranging between 47.77% for **amb-class** to 76.50% for **rdm**, and with only one model, **amb-dim** below 50% (see also Table C.3). While struggling on the positive classes - in a way that is similar to models fine-tuned on abusive language only - the pattern on the non-hateful tests indicates that the presence of an extra dimension (i.e., offensive language) seems to improve the overall precision. Although the behaviour on the DALC-v2.0 held-out test may suggest that this could be due by chance rather than robustness, the performance on the challenging functionalities **F9–F11** cautiously indicates the contrary. Indeed, this is the only case where only one fine-tuned model has performance below 50% (**amb-class** for both tests). For **F11**, the best accuracy (70.00% - **amb-dim**) is better than that of the models trained on abusive language only. Further improvements can be seen for **F21** with two models above 50% (**amb-dim** and **rdm**), and **F24**, with three models (**std**, **amb-dim** and **rdm**). At the same time, issues persist on other functionalities. In particular, for **F23** we observe a downgrade of the accuracy when compared to the abusive language models, and for **F16**, where all models are well below the 50% threshold.

A notable difference, when compared to DALC-v2.0, concerns the behaviour of the data maps training splits. With the sole exception of the **amb-dim** from the offensive dimension, in all the other cases they help to achieve better results when compared to the use of the full training set as well as the use of random training splits. In particular, the selection of ambiguous data per dimension (**amb-dim**) consistently outperforms all other settings, a trend already observed for DALC-v2.0. Although for the abusive dimension we observe a better results for the **amb-dim** setting, the difference is not statistically significant.

Focusing on the best models, the use of offensive data allows the model to achieve 85.50% accuracy on all hateful tests on average, while it only obtains 76.88% with abusive data and 72.64% for the joint model. In only two functionalities, namely **F5** (direct threat) and **F7** (hateful slurs), the use of abusive language obtains better results. As for the joint model, the best results are mainly on the non-hateful functionalities, namely **F19** (use of protected group identifiers in a positive statement), **F20** (denouncement of hate via quote) and **F22** (abuse at objects). The only hateful functionality where it obtains the best score is **F26** (change of hateful term by eliminating characters).

Finally, it is clear that the annotations in DALC-v2.0, and consequently the fine-tuned models, have limits that emerge with HATECHEK-NL while being hidden by looking at their performances of the respective DALC-v2.0 test sets. Even the use of abusive language data, which are the most similar to hate speech to fine-tune models, does not allow to properly pass all the tests. From the analysis of the results of every single functional test, it appears evident that very good results are obtained on the easy cases: as soon the expressions of hate become more subtle or fine-grained, models fine-tuned on DALC-v2.0, regardless of the training split and annotated dimension used, fail.

OP-NL Results for OP-NL are also reported on Table 5. Differently from HATECHEK-NL, we have converted the prediction for the ABU class of the joint model into offensive labels.

Like in the previous cases, all fine-tuned models outperform the baselines. The use of the full training data (**std**) results in the best scores only for the offensive and the joint models, while the model fine-tuned on abusive language only underperforms. This is actually a positive result: abusive language is more specific than its offensive counterpart, and the lower results further confirm the quality of the annotated data for each language phenomenon in DALC-v2.0. On the other hand, the results for the joint model are quite disappointing. Although competitive with the offensive dimension model, the results are ≈ 2 points lower. By looking at the distribution of the errors, we observe that the biggest sources of errors are offensive messages misclassified as NOT, a behaviour in-line with what we have observed when the same model is evaluated against the DALC-v2.0 held-out test set.

Similarly to the other evaluation settings, the **amd-class** data maps for the offensive and abusive models in isolation obtain competitive results when compared to the **std** models. When using the abusive language dimension as training material, the model fine-tuned with **amd-class** achieves the best macro F1 (68.75). Only for the joint model, we observe better results for the **rdm** splits. Lastly, the only model which across all training splits overgeneralises the positive class is the joint model. On the basis of the errors observed in DALC-v2.0 for this model, it appears that the overgeneralisation is a consequence of the conversion process of the labels for offensiveness to make the predictions compatible with OP-NL.

5 Discussion

Concerning data maps, we observe inconsistent behaviours of the fine-tuned models: on DALC-v2.0, they are unsuccessful while they achieve either the best performances or very competitive results on HATECHEK-NL and OP-NL. By analysing the variability per class across **amb-dim**, **amb-class**, and **rdm**, we can see that **amb-dim** is the data split that contains core ambiguous cases for all classes, separately and jointly. The ambiguity for the positive class remain relatively high also in **amb-class**, but we observe a drop in the values for the NOT class (0.096 for offensive language, 0.062 for abusive language, and 0.095 when the two dimensions jointly). This means that in the negative class we mainly have easy examples and relatively ambiguous cases for the positive classes. A similar distribution can be observed for the variability for all **rdm** splits, where the variability for the negative class is substantially lower than that of the positive classes. When compared to our expectations on the behaviour of the models based on the ambiguous and the random splits, these observations help to explain the results of these models. Overall, the use of ambiguous examples only on the positive class(es) forces models to pay more attention towards the challenging cases and "disregard" the contributions of the easy ones. This confirms our explanation for the overgeneralisation of the positive class(es). As for the randomly extracted data (**rdm**), it appears that their better performances on DALC-v2.0 is an effect of the distribution of the training instances closer to those in the held-out test data. As for the **amb-dim**, there is a consistent pattern of underperformance across all test data. Rather than issues in the variability scores, i.e., not very "strong" ambiguous cases, it appears that the culprit for the low results should be found in the size of the original DALC-v2.0 training data which makes it difficult to identify good ambiguous cases with respect to the easy (or hard) ones. A similar pattern has been identified by Richburg and Carpuat (2022) when applying data cartography to low- and very-low Machine Translation settings. Furthermore, across all the test sets, we found that only for HATECHEK-NL the use of ambiguous training instances leads to improved out-of-domain performance as reported by Swayamdipta et al. (2020).

When comparing the results of our models against the English HATECHECK for a BERT model fine-tuned on Davidson et al. (2017), the core set of non-hateful functional tests (i.e., **F9**, **F20–21**, **F23–24**) are consistently failed in both languages. Things are quite different for MHC. In this case, the tested model is fine-tuned by concatenating three datasets whose definitions of hate speech perfectly matches the one adopted in MHC. While for **F9** results are excellent, the model still struggles for **F20–21**, **F23–24**[2]

6 Conclusions and Future Directions

In this paper we have presented an extensive benchmarking of models fine-tuned with DALC-v2.0 across three test portions: an internal held-out test, a functional benchmark, HATECHEK-NL, and a dynamic test, OP-NL. Our experiments have investigated the reliability of DALC-v2.0 as a training set for three classification tasks: offensive and abusive language detection in isolation and jointly. Overall, addressing each task in isolation results in better performances than when running a joint experiment. The challenge here lies both in the strict connections between the two language phenomena in analysis and in the limited training data. When the fine-tuned models are applied on the out-of-corpus test sets, we observe a good performance on OP-NL and less satisfying results on HATECHEK-NL. The compatibility of the annotated phenomena in the training data actually plays a major role on this behaviour and it indicates that the quality of the annotated data in DALC-v2.0 contributes to develop robust models.

We have further investigated the effectiveness of the use of data cartography to identify more informative subsets of training materials. Unlike previous work, we observe a limited beneficial effects of this data selection method with DALC-v2.0. While the size of the dataset appears limited for an effective application of this method, we have found that selecting training subsets on the basis of the training dynamics of each annotated dimension results in better systems than when using training dynamics of the whole training split.

The results on HATECHEK-NL clearly identify limitations of the use of DALC-v2.0 to detect hate speech. While its abusive dimension can be considered a good proxy, all fine-tuned models systematically fails on core non-hateful functional tests, indicating limitations in the annotated data.

Future work will focus on extending DALC-v2.0 with multiple hate speech datasets and further

[2] These correspond to **F18–19**, **F21–22** in MHC.

validate the functionalities of HATECHEK-NL.

Ethical statement

Limitations HATECHECK-NL is based on MHC and it inherits its limits. However, as we have discussed in Section 2, we failed to fully implement some functional tests (e.g., reappropriation of slurs) because we were not able to find evidence during our research. To address these limitations, we plan to conduct focused interviews with Dutch organizations such as The Black Archives[3].

Intended use HATECHEK-NLis a diagnostic tool for hate speech against specific protected groups. We have shown its functionalities and its impact on the evaluation of models trained both on a different language phenomenon, e.g., offensive language, and on related and comparable one, e.g., abusive language. The results have shown critical weaknesses mainly on the non-hateful tests rather than showing the strengths of the systems/models on the hateful examples. Similarly, OP-NL is a dynamic test for offensive language whose use is to help assessing the robustness and portability of models trained for offensive language detection.

Goodness of data DALC-V2.0 is the only publicly available resource for investigating the behavior of models on offensive and abusive language phenomena in Dutch. None of the annotated dimensions in DALC-V2.0 explicitly address hate speech as we discussed in Section 2. The results of the fine-tuned models on HATECHEK-NL for the abusive language dimension indicate a compatibility between abusive language in DALC-V2.0 and hate speech. The use of offensive training data on HATECHEK-NL better highlights the limitations of the data, especially as pointed out by the systematic failure on the functions **F23–24**. At the same time, the results on OP-NL for offensive language show a relatively good portability of the models for this language phenomenon.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. 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*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in NLI: Ways (not) to go beyond simple heuristics. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tommaso Caselli, Arjan Schelhaas, Marieke Weultjes, Folkert Leistra, Hylke van der Veen, Gerben Timmerman, and Malvina Nissim. 2021. DALC: the Dutch abusive language corpus. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 54–66, Online. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 512–515.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.

Heather Lent, Semih Yavuz, Tao Yu, Tong Niu, Yingbo Zhou, Dragomir Radev, and Xi Victoria Lin. 2021. Testing cross-database semantic parsers with canonical utterances. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 73–83, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marta Marchiori Manerba and Sara Tonelli. 2021. Fine-grained fairness analysis of abusive language detection systems with CheckList. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 81–91, Online. Association for Computational Linguistics.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, pages 1–47.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[3] `https://www.theblackarchives.nl/over-ons.html`

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Aquia Richburg and Marine Carpuat. 2022. Data cartography for low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5594–5607, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Paul Röttger, Haitham Seelawi, Debora Nozza, Zeerak Talat, and Bertie Vidgen. 2022. Multilingual HateCheck: Functional tests for multilingual hate speech detection models. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 154–169, Seattle, Washington (Hybrid). Association for Computational Linguistics.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online. Association for Computational Linguistics.

Ward Ruitenbeek, Victor Zwart, Robin Van Der Noord, Zhenja Gnezdilov, and Tommaso Caselli. 2022. "zo grof !": A comprehensive corpus for offensive and abusive language in Dutch. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 40–56, Seattle, Washington (Hybrid). Association for Computational Linguistics.

Ananya B. Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M. Khapra. 2021. Perturbation CheckLists for evaluating NLG evaluation metrics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7219–7234, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Dion Theodoridis and Tommaso Caselli. 2022. All that glitters is not gold: Transfer-learning for offensive language detection in dutch. *Computational Linguistics in the Netherlands Journal*, 12:141–164.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 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, Volume 1 (Long and Short Papers)*, pages 602–608, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

A HateCheck-NL: List of Functional Tests

	Functionality	Description from Röttger et al. (2021)	Label	Count	
				templ	cases
F1	derog_neg_emote_h	Strong negative emotions explicitly expressed about a protected group or its members	hateful	20	140
F2	derog_neg_attrib_h	Explicit descriptions of a protected group or its members using very negative attributes	hateful	20	140
F3	derog_dehum_h	Explicit dehumanisation of a protected group or its members	hateful	20	140
F4	derog_impl_h	Implicit derogation of a protected group or its members	hateful	20	140
F5	threat_dir_h	Direct threats against a protected group or its members	hateful	20	140
F6	threat_norm_h	Threats expressed as normative statements	hateful	20	140
F7	slur_h	Hate expressed using slurs	hateful	10	170
F8	slur_homonym_nh	Non-hateful homonyms of slurs	non-hate	25	25
F9	slur_reclaimed_nh	Use of reclaimed slurs	non-hate	45	45
F10	profanity_h	Hate expressed using profanity	hateful	20	140
F11	profanity_nh	Non-hateful uses of profanity	non-hate	100	100
F12	ref_subs_clause_h	Hate expressed through pronoun reference in subsequent clauses	hateful	20	140
F13	ref_subs_sent_h	Hate expressed through pronoun reference in subsequent sentences	hateful	20	140
F14	negate_pos_h	Hate expressed using negated positive statements	hateful	20	140
F15	negate_neg_nh	Non-hate expressed using negated hateful statements	non-hate	20	140
F16	phrase_question_h	Hate phrased as a question	hateful	20	140
F17	phrase_opinion_h	Hate phrased as an opinion	hateful	20	140
F18	ident_neutral_nh	Neutral statements using protected group identifiers	non-hate	20	140
F19	ident_pos_nh	Positive statements using protected group identifiers	non-hate	30	210
F20	counter_quote_nh	Denouncements of hate that quote it	non-hate	20	170
F21	counter_ref_nh	Denouncements of hate that make direct reference to it	non-hate	20	170
F22	target_obj_nh	Abuse targeted at objects	non-hate	65	65
F23	target_indiv_nh	Abuse targeted at individuals not referencing membership in a protected group	non-hate	65	65
F24	target_group_nh	Abuse targeted at non-protected groups (e.g. professions)	non-hate	65	65
F25	spell_char_swap_h	Swaps of adjacent characters	hateful	20	140
F26	spell_char_del_h	Missing characters	hateful	20	140
F27	spell_space_del_h	Missing word boundaries	hateful	20	170
F28	spell_space_add_h	Added spaces between characters	hateful	20	170
F29	spell_leet_h	Leet speak	hateful	20	170
	Total		**hateful**	**350**	**2,640**
			non-hate	**475**	**1,195**
			all	**825**	**3,835**

Table A.1: HATECHECK-NL functionality overview

B Replicability: Preprocessing and Hyperparameters

Preprocessing All experiments have been conducted with common pre-processing steps, namely:

- lowercasing of all words
- all users' mentions have been substituted with a placeholder (MENTION);
- all URLs have been substituted with a with a placeholder (URL);
- all ordinal numbers have been replaced with a placeholder (NUMBER);
- emojis have been replaced with text (e.g. 😹 → `:cat_face_joy:`) using Python `emoji` package;
- hashtag symbol has been removed from hasthtags (e.g. #kadiricinadalet → kadiricinadalet);
- extra blank spaces have been replaced with a single space;
- extra blank new lines have been removed.

Models' hyperparameters All hyperparamters used for the experiments are reported in Table B.1.

Model	Task	Hyperparm.	Value
BERTje	Offensive Abusive Offensive & Abusive	Learning rate	2e-5
		Training Epochs	5
		Optimzer	AdamW
		Adam epsilon	1e-8
		Max sequence length	280
		Batch size	16
		Num. warmup steps	2

Table B.1: Hyperparameters used to fine-tune BERTje.

C Detailed Results

System	Train	Class	P	R	Macro-F1
Dummy	n.a.	OFF	0.0	0.0	0.4230
		NOT	0.7340	1.0	
BERTje	std	OFF	0.7214	0.6864	0.7993
		NOT	0.8881	0.9047	
	amb-dim	OFF	0.5031	0.6459	0.6885
		NOT	0.8577	0.7699	
	amb-class	OFF	0.6575	0.6932	0.7766
		NOT	0.8871	0.8697	
	rand	OFF	0.7139	0.6294	0.7764
		NOT	0.8723	0.9064	

Table C.1: DALC-v2.0 **offensive language**: binary classification; **rand** reports the averages of the results obtained using three different training splits.

Model	Train	Class	P	R	Macro-F1
Dummy	n.a.	ABU	0.0	0.0	0.4619
		NOT	0.8584	1.0	
BERTje	std	ABU	0.5741	0.4687	0.7223
		NOT	0.9149	0.9426	
	amb-dim	ABU	0.3783	0.5270	0.6631
		NOT	0.9166	0.8571	
	amb-class	ABU	0.3693	0.7106	0.6721
		NOT	0.9434	0.7852	
	rand	ABU	0.5534	0.4527	0.7070
		NOT	0.9104	0.9417	

Table C.2: DALC-v2.0 **abusive language**: binary classification; **rand** reports the averages of the results obtained using three different training splits.

Model	Train	Class	P	R	Macro-F1
Dummy	n.a.	OFF	0.0	0.0	0.2824
		ABU	0.0	0.0	
		NOT	0.7348	1.0	
BERTje	std	OFF	0.3301	0.3391	0.5890
		ABU	0.5696	0.5011	
		NOT	0.8971	0.9800	
	amb-dim	OFF	0.1933	0.4158	0.4374
		ABU	0.2718	0.4773	
		NOT	0.8822	0.5830	
	amb-class	OFF	0.2194	0.4653	0.5358
		ABU	0.4491	0.5529	
		NOT	0.9371	0.7187	
	rand	OFF	0.3343	0.2953	0.5725
		ABU	0.5778	0.4672	
		NOT	0.8682	0.9159	

Table C.3: DALC-v2.0 **offensive and abusive language**: multi-class classification; **rand** reports the averages of the results obtained using three different training splits.

	Functionality	Label	# Inst.	std	amb-dim	amb-class	rdm
F1	derog_neg_emote_h	hateful	140	77.10	61.40	**93.60**	69.77
F2	derog_neg_attrib_h	hateful	140	85.00	95.00	**98.60**	87.37
F3	derog_dehum_h	hateful	140	78.60	**91.40**	85.70	69.53
F4	derog_impl_h	hateful	140	37.10	**65.70**	56.40	31.63
F5	threat_dir_h	hateful	140	58.60	57.90	**77.90**	47.87
F6	threat_norm_h	hateful	140	57.90	78.60	**88.60**	53.80
F7	slur_h	hateful	170	71.20	**90.60**	79.40	67.47
F8	slur_homonym_nh	non-hate	25	68.00	40.00	64.00	**73.33**
F9	slur_reclaimed_nh	non-hate	45	46.70	33.30	26.70	49.63
F10	profanity_h	hateful	140	**98.60**	93.60	**98.60**	97.60
F11	profanity_nh	non-hate	100	29.00	15.00	19.00	33.67
F12	ref_subs_clause_h	hateful	140	75.00	85.00	**98.60**	73.80
F13	ref_subs_sent_h	hateful	140	88.60	95.70	**99.30**	85.27
F14	negate_pos_h	hateful	140	40.70	65.70	**77.10**	31.17
F15	negate_neg_nh	non-hate	140	65.70	50.70	12.90	**65.93**
F16	phrase_question_h	hateful	140	52.90	11.40	**69.30**	49.50
F17	phrase_opinion_h	hateful	140	67.90	65.70	**82.10**	55.50
F18	ident_neutral_nh	non-hate	140	83.60	42.90	69.30	**91.47**
F19	ident_pos_nh	non-hate	210	65.20	57.60	40.50	**73.80**
F20	counter_quote_nh	non-hate	170	38.20	37.10	28.20	**50.77**
F21	counter_ref_nh	non-hate	170	27.10	14.10	11.80	31.73
F22	target_obj_nh	non-hate	65	61.50	15.40	38.50	**64.63**
F23	target_indiv_nh	non-hate	65	41.50	18.50	12.30	46.67
F24	target_group_nh	non-hate	65	26.20	26.20	12.30	30.27
F25	spell_char_swap_h	hateful	140	57.10	68.60	**82.10**	60.93
F26	spell_char_del_h	hateful	140	72.10	**89.30**	87.10	76.20
F27	spell_space_del_h	hateful	170	82.90	84.70	**95.90**	86.07
F28	spell_space_add_h	hateful	170	55.90	**78.80**	78.20	42.77
F29	spell_leet_h	hateful	170	70.60	**91.20**	87.10	72.37
	Average			61.40	59.35	**64.52**	61.05
	Average - Hateful			68.86	76.57	**85.50**	64.57
	Average - Non-hateful			47.61	30.53	28.77	52.57

Table C.1: HATECHEK-NL: results using training data from DALC-v2.0 annotated for **offensive language**. Best results across training splits are marked in bold. We have marked in red results below 50%.

	Functionality	Label	# Inst.	std	amb-dim	amb-class	rdm
F1	derog_neg_emote_h	hateful	140	57.10	**69.30**	64.30	48.33
F2	derog_neg_attrib_h	hateful	140	77.10	**93.60**	83.60	65.00
F3	derog_dehum_h	hateful	140	61.40	**80.00**	**80.00**	53.10
F4	derog_impl_h	hateful	140	35.70	**55.00**	27.90	24.53
F5	threat_dir_h	hateful	140	65.70	**86.40**	67.10	56.20
F6	threat_norm_h	hateful	140	61.40	**80.00**	70.00	43.33
F7	slur_h	hateful	170	63.50	**91.20**	78.20	44.10
F8	slur_homonym_nh	non-hate	25	**80.00**	32.00	48.00	78.67
F9	slur_reclaimed_nh	non-hate	45	**66.70**	44.40	48.90	58.53
F10	profanity_h	hateful	140	85.00	**95.70**	**95.70**	79.27
F11	profanity_nh	non-hate	100	50.00	29.00	34.00	**62.67**
F12	ref_subs_clause_h	hateful	140	73.60	80.00	**80.70**	53.83
F13	ref_subs_sent_h	hateful	140	84.30	86.40	**94.30**	69.53
F14	negate_pos_h	hateful	140	36.40	**67.90**	49.30	20.00
F15	negate_neg_nh	non-hate	140	67.90	49.30	60.70	**74.77**
F16	phrase_question_h	hateful	140	24.30	14.30	30.00	11.90
F17	phrase_opinion_h	hateful	140	57.90	**77.90**	54.30	25.23
F18	ident_neutral_nh	non-hate	140	85.00	61.40	80.70	**91.20**
F19	ident_pos_nh	non-hate	210	63.30	35.20	62.40	**81.90**
F20	counter_quote_nh	non-hate	170	47.10	52.90	59.40	**76.87**
F21	counter_ref_nh	non-hate	170	48.80	39.40	37.10	**59.40**
F22	target_obj_nh	non-hate	65	86.20	52.30	70.80	**93.30**
F23	target_indiv_nh	non-hate	65	43.10	13.80	33.80	**51.80**
F24	target_group_nh	non-hate	65	43.10	18.50	27.70	**56.43**
F25	spell_char_swap_h	hateful	140	51.40	**83.60**	71.40	40.70
F26	spell_char_del_h	hateful	140	60.70	82.90	**84.30**	51.43
F27	spell_space_del_h	hateful	170	79.40	87.60	**92.40**	55.67
F28	spell_space_add_h	hateful	170	33.50	**74.10**	47.10	30.40
F29	spell_leet_h	hateful	170	55.90	**84.70**	76.50	45.07
	Average			60.19	**62.72**	62.43	55.28
	Average - Hateful			59.58	**76.88**	69.16	45.70
	Average - Non-hateful			57.38	36.29	48.14	**65.72**

Table C.2: HateCheck-NL: results using training data from DALC-v2.0 annotated for **abusive language**. Best results across training splits are marked in bold. We have marked in red results below 50%.

	Functionality	**Label**	**# Inst.**	**std**	**amb-dim**	**amb-class**	**rdm**
F1	derog_neg_emote_h	hateful	140	59.30	63.60	**77.90**	30.27
F2	derog_neg_attrib_h	hateful	140	77.10	65.70	**83.60**	50.27
F3	derog_dehum_h	hateful	140	62.90	77.90	**78.60**	49.30
F4	derog_impl_h	hateful	140	31.40	**50.00**	49.30	19.27
F5	threat_dir_h	hateful	140	57.10	77.10	**80.70**	41.20
F6	threat_norm_h	hateful	140	57.10	59.30	**67.10**	34.03
F7	slur_h	hateful	170	64.70	72.40	**77.60**	46.07
F8	slur_homonym_nh	non-hate	25	80.00	64.00	56.00	**82.67**
F9	slur_reclaimed_nh	non-hate	45	51.10	**62.20**	35.60	61.47
F10	profanity_h	hateful	140	88.60	72.10	**91.40**	70.23
F11	profanity_nh	non-hate	100	55.00	**70.00**	40.00	66.00
F12	ref_subs_clause_h	hateful	140	75.00	76.40	**80.00**	46.90
F13	ref_subs_sent_h	hateful	140	82.10	87.10	**90.70**	63.80
F14	negate_pos_h	hateful	140	56.40	**67.90**	17.87	20.00
F15	negate_neg_nh	non-hate	140	75.00	60.70	50.00	**85.93**
F16	phrase_question_h	hateful	140	32.90	25.00	21.40	11.20
F17	phrase_opinion_h	hateful	140	49.30	41.40	**60.70**	21.90
F18	ident_neutral_nh	non-hate	140	80.70	46.40	67.90	**89.77**
F19	ident_pos_nh	non-hate	210	65.20	39.00	53.80	**83.17**
F20	counter_quote_nh	non-hate	170	62.40	**84.40**	64.10	84.13
F21	counter_ref_nh	non-hate	170	48.20	50.60	36.50	**69.40**
F22	target_obj_nh	non-hate	65	87.70	86.20	75.40	**92.30**
F23	target_indiv_nh	non-hate	65	36.90	27.70	15.40	**57.43**
F24	target_group_nh	non-hate	65	61.50	56.90	30.80	**69.23**
F25	spell_char_swap_h	hateful	140	46.40	58.60	**72.90**	31.90
F26	spell_char_del_h	hateful	140	66.40	62.10	**84.30**	47.37
F27	spell_space_del_h	hateful	170	73.50	65.90	**85.90**	48.07
F28	spell_space_add_h	hateful	170	42.40	45.90	**75.90**	21.57
F29	spell_leet_h	hateful	170	58.20	72.40	**75.30**	37.83
	Average			60.94	61.22	**63.21**	52.78
	Average - Hateful			59.09	62.74	**72.64**	38.28
	Average - Non-hateful			63.97	58.74	47.77	**76.50**

Table C.3: HATECHEK-NL: results using training data from DALC-v2.0 annotated for **offensive and abusive language**. Best results across training splits are marked in bold. We have marked in red results below 50%.

System	Train	Class	P	R	Macro-F1
Dummy	n.a.	OFF	0.0	0.0	0.3904
		NOT	0.6406	1.0	
BERTje	std	OFF	0.6772	0.6345	0.7356
		NOT	0.8020	0.8304	
	amb-dim	OFF	0.4293	0.8219	0.5423
		NOT	0.7949	0.3871	
	amb-class	OFF	0.6527	0.5510	0.6991
		NOT	0.7684	0.8356	
	rand	OFF	0.6761	0.5028	0.6907
		NOT	0.7562	0.8625	

Table C.4: OP-NL **offensive language**: binary classification; **rand** reports the averages of the results obtained using three different training splits.

Model	Train	Class	P	R	Macro-F1
Dummy	n.a.	OFF	0.0	0.0	0.3904
		NOT	0.6406	1.0	
BERTje	std	OFF	0.8582	0.2134	0.5757
		NOT	0.6896	0.9802	
	amb-dim	OFF	0.6773	0.3544	0.6319
		NOT	0.7143	0.9053	
	amb-class	OFF	0.6446	0.5250	0.6875
		NOT	0.7587	0.8377	
	rand	OFF	0.8217	0.1911	0.5500
		NOT	0.6829	0.9761	

Table C.5: OP-NL **abusive language**: binary classification; **rand** reports the averages of the results obtained using three different training splits.

Model	Train	Class	P	R	Macro-F1
Dummy	n.a.	OFF	0.0	0.0	0.3904
		NOT	0.6406	1.0	
BERTje	std	OFF	0.6606	0.6030	0.7185
		NOT	0.7877	0.8262	
	amb-dim	OFF	0.4002	0.6809	0.5183
		NOT	0.7050	0.4277	
	amb-class	OFF	0.5278	0.7570	0.6641
		NOT	0.8198	0.6202	
	rand	OFF	0.7045	0.4990	0.6991
		NOT	0.7591	0.8824	

Table C.6: OP-NL **offensive and abusive language**: binary classification; **rand** reports the averages of the results obtained using three different training splits.