A Neighborhood Framework for Resource-Lean Content Flagging

Sheikh Muhammad Sarwar[2,5,*] **and Dimitrina Zlatkova**[1] **and Momchil Hardalov**[1,6] **and Yoan Dinkov**[1] **and Isabelle Augenstein**[1,3] **and Preslav Nakov**[1,4]

[1]Checkstep, UK, [2]University of Massachusetts, Amherst, [3]University of Copenhagen, Denmark,
[4]Qatar Computing Research Institute, HBKU, Qatar [5]Amazon.com, US
[6]Sofia University ''St. Kliment Ohridski'', Bulgaria
smsarwar@amazon.com,
{didi,momchil,yoan.dinkov,isabelle,preslav.nakov}@checkstep.com

Abstract

We propose a novel framework for cross-lingual content flagging with limited target-language data, which significantly outperforms prior work in terms of predictive performance. The framework is based on a nearest-neighbor architecture. It is a modern instantiation of the vanilla k-nearest neighbor model, as we use Transformer representations in all its components. Our framework can adapt to new source-language instances, without the need to be retrained from scratch. Unlike prior work on neighborhood-based approaches, we encode the neighborhood information based on query–neighbor interactions. We propose two encoding schemes and we show their effectiveness using both qualitative and quantitative analysis. Our evaluation results on eight languages from two different datasets for abusive language detection show sizable improvements of up to 9.5 F1 points absolute (for Italian) over strong baselines. On average, we achieve 3.6 absolute F1 points of improvement for the three languages in the Jigsaw Multilingual dataset and 2.14 points for the WUL dataset.

1 Introduction

Online content moderation is an increasingly important problem—small-scale websites and large-scale corporations alike strive to remove harmful content from their platforms (Vidgen et al., 2019; Pavlopoulos et al., 2017; Wulczyn et al., 2017). This is partly in anticipation of proposed legislation, such as the *Digital Service Act* (European Commission, 2020) in the EU and the *Online Harms Bill* (UK Government, 2020) in the UK. Moreover, the lack of content moderation can have significant impact on businesses (e.g., Parler was denied server space), on governments (e.g., the U.S. Capitol Riots), and on individuals (e.g., because hate speech is linked to self-harm [Jürgens et al., 2019]).

A key challenge when developing content moderation systems is the lack of resources for many languages (other than English). With this in mind, here we aim to create a content flagging model for a target language with limited annotated data by transferring knowledge from another dataset in a different language, for which a large amount of training data is available.

Various approaches have been proposed in the literature to address the lack of enough training data in the target language. A popular approach is to fine-tune large-scale pre-trained multilingual language models such as XLM (Conneau and Lample, 2019), XLM-R (Conneau et al., 2020), or mBERT (Devlin et al., 2019) on the target dataset (Glavaš et al., 2020; Stappen et al., 2020). In order to incorporate knowledge from the source dataset, a sequential adaptation technique can be used that first fine-tunes a multilingual language model (LM) on the source dataset, and then on the target dataset (Garg et al., 2020). There are also existing approaches for mixing the source and the target datasets (Shnarch et al., 2018) in different proportions, followed by fine-tuning the multilingual language model on the resulting dataset. While sequential adaptation introduces the risk of forgetting the knowledge from the source dataset, such mixing methods are driven by heuristics that are effective, but not systematic. Crucially, as we argue in this paper, this is because they do not model the relationship between the source and the target datasets. Another problem arises if we consider that examples with novel labels can be added to the source dataset. This is a specifically pertinent issue for content moderation, as efforts to create new resources often lead to the introduction of new label inventories or taxonomies (Banko et al.,

[*]Work done prior to joining Amazon

Transactions of the Association for Computational Linguistics, vol. 10, pp. 484–502, 2022. https://doi.org/10.1162/tacl_a_00472
Action Editor: Bo Pang. Submission batch: 5/2021; Revision batch: 12/2021; Published 5/2022.

2020). In that case, model re-training becomes a requirement in order to be able to map the new label space to the output layer that is used for fine-tuning.

We propose a Transformer-based k-nearest neighbor (kNN$^+$) framework,[1] a one-stop solution and a significant improvement over the vanilla kNN model. Our framework addresses the above-mentioned challenges, which are not easy to solve via simple fine-tuning of pre-trained language models. Moreover, to the best of our knowledge, our framework is the first attempt to use kNN for transfer learning for the task of abusive content detection.

Given a query, which is a training or an evaluation data point from the target dataset, kNN$^+$ retrieves its nearest neighbors using a language-agnostic sentence embedding model. Then, it constructs Transformer representations for the query and for its neighbors. After that, it computes interaction features, which are based on the interactions of the representations of the query with each of its neighbors.[2] At training time, the interaction features are optimized using supervised training signals computed from the label of the query and the neighbor, so that the features indicate their level of agreement.

For example, if the query and its neighbor are both abusive, they agree on the labels. Thus, the interactions help the model learn a semantic similarity space in terms of labels. The framework further uses a self-attention mechanism to aggregate the interaction features from all the neighbors, and it uses the aggregated representation to classify the input query. This representation is computed from the interaction features and indicates the agreement of the query with the neighborhood. As the predictions are made based on aggregated interaction features only, kNN$^+$ can easily incorporate new examples with unseen labels without requiring re-training. The conceptual framework is shown in Figure 1; It is robust to neighbors with incorrect labels, as it can learn to disagree with them as part of its training process.

We instantiate two variants of our framework: Cross-Encoder (CE) kNN$^+$ and Bi-Encoder (BE) kNN$^+$. The CE kNN$^+$ concatenates the query and a neighbor, and passes that sequence through a Transformer to obtain interaction features. BE kNN$^+$ computes representations of the query and of a neighbor by passing them individually through a Transformer, and computes interaction features from these representations. BE kNN$^+$ is more efficient than CE kNN$^+$, but it does not yield the same performance gains. Both models outperform six strong baselines both in cross-lingual and in multilingual settings. Our contributions can be summarized as follows:

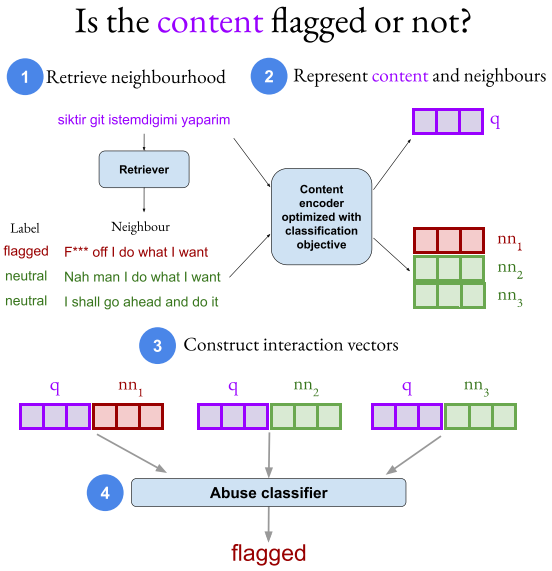


Figure 1: Conceptual diagram of our neighborhood framework. The query is processed using run-time compute, while the neighbor vector is pre-computed.

- We address cross-lingual transfer learning for content flagging with limited labeled data from the target language.
- We demonstrate that neighborhood methods, such as kNNs, are viable candidates for approaching content flagging.
- We propose a novel framework, kNN$^+$, which, unlike a vanilla kNN, models the relationship between a data point and each of its neighbors to represent the neighborhood, using language-agnostic Transformers.
- Our evaluation results on eight languages from two different datasets for abusive language detection show sizable improvements of up to 9.5 F1 points absolute (for Italian) over strong baselines. On average, we

[1]We use a '+' superscript to indicate that our kNN$^+$ framework is an improvement over the vanilla kNN model.

[2]We borrow the terminology from information retrieval, as the interactions between a query and a document in deep matching models are computed in a similar way (Guo et al., 2016).

achieve improvements of 3.6 F1 points for the three languages in the Jigsaw Multilingual dataset, and of 2.14 F1 points on the WUL dataset.

2 Related Work

Below, we review recent work on abusive language detection and neighborhood approaches.

2.1 Abusive Content Detection

Most approaches for abusive language detection use text classification models, which have been shown to be effective for related tasks such as sentiment analysis. This includes SVMs (MacAvaney et al., 2019), CNNs (Georgakopoulos et al., 2018; Badjatiya et al., 2019; Agrawal and Awekar, 2018), LSTMs (Arango et al., 2019; Agrawal and Awekar, 2018), BiLSTMs, with attention (Agrawal and Awekar, 2018), Capsule networks (Srivastava et al., 2018), and fine-tuned Transformers (Glavaš et al., 2020). All these approaches focus on single data points, while we also model their neighbourhoods. See Nakov et al. (2021) for a recent survey of abusive language detection.

Several papers studied the bias in hate speech detection datasets and criticized the use of within-dataset evaluations (Arango et al., 2019; Davidson et al., 2019; Badjatiya et al., 2019), as this is not a realistic setting, and findings about generalizability based on such experimental settings are questionable. A more realistic and robust evaluation setting was investigated by Glavaš et al. (2020), who showed the performance of online abuse detectors in a zero-shot cross-lingual setting. They fine-tuned several multilingual language models (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020; Sanh et al., 2019; Wang et al., 2020) such as XLM-RoBERTa and mBERT on English datasets and observed how these models transfer to datasets in five other languages. Other cross-lingual abuse detection efforts include using Twitter user features for detecting hate speech in English, German, and Portuguese (Fehn Unsvåg and Gambäck, 2018); cross-lingual embeddings (Ranasinghe and Zampieri, 2020); and using multingual lexicon with deep learning (Pamungkas and Patti, 2019). Considerable relevant research was also done as part of the OffensEval shared task at SemEval (Zampieri et al., 2019a,b, 2020; Rosenthal et al., 2021).

While understanding the performance of zero-shot cross-lingual models is interesting from a natural language understanding point of view, in reality, a platform willing to deploy an abusive language detection system can almost always provide some examples of malicious content for training.

Thus, a few-shot or a low-shot scenario is more realistic, and we approach cross-lingual transfer learning from that perspective. We hypothesize that a nearest-neighbor model is a reasonable choice in such a scenario, and we propose several improvements over such a model.

2.2 Neighbourhood Models

kNN models have been used for a number of NLP tasks such as part of speech tagging (Daelemans et al., 1996) and morphological analysis (Bosch et al., 2007), among many others. Their effectiveness is rooted in the underlying similarity function, and thus non-linear models such as neural networks can bring additional boost to their performance. More recently, Kaiser et al. (2017) used a similarly differentiable memory that is learned and updated during training and is then applied to one-shot learning tasks. Khandelwal et al. (2020) introduced kNN retrieval for improving language modeling, which Kassner and Schütze (2020) extended to question answering (QA). Guu et al. (2020) proposed a framework for retrieval-augmented language modeling (REALM), showing its effectiveness on three Open QA datasets. Lewis et al. (2020) explored a retrieval-augmented generation for a variety of tasks, including fact-checking and QA, among others. Fan et al. (2021) introduced a kNN framework for dialogue generation using pre-trained embeddings enhanced by learning an alignment function for retrieval from a set of external multi-modal evidence sources. Finally, Wallace et al. (2018) proposed a deep kNN approach for interpreting the predictions from a neural network for the task of natural language inference.

All the above approaches use neighbors as additional information sources, but do not consider the interactions between the neighbors as we do. Moreover, there is no existing work on using deep kNN models for cross-lingual abusive content detection.

3 kNN$^+$ Framework

We present our kNN$^+$ framework below.

3.1 Problem Setting

Our goal is to learn a content flagging model from source and target datasets in different languages with different label spaces—see Figure 1 for an illustration of our framework.

Formally, we assume access to a source dataset for content flagging, $D^s = \{(x_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$, where x_i^s is a textual content and $\mathbf{y}_i^s \in \mathcal{Y}$. Further, a target dataset is given, $D^t = \{(x_j^t, y_j^t)\}_{j=1}^{n_t}$, where $y_j^t \in \{flagged, neutral\}$. D^s is resource-rich (i.e., $n_s \gg n_t$) and label-rich (i.e., $|\mathcal{Y}| > 2$). The label space, $\mathcal{Y} = \{hate, insult, \ldots, neutral\}$, of D^s contains fine-grained labels for different levels of abusiveness along with the *neutral* label. We convert the label space of D^s to align it with the label space of D^t as follows: $\mathcal{Y}' = \{flagged \mid x \in \mathcal{Y}, x \neq neutral\}$. Note that this conversion is needed at training time to compute label agreement in our proposed neighborhood framework. However, at inference time, a conversion of the label space of D^s is not needed, as the label of an item from D^t is predicted using the latent representations of the neighbors, rather than their labels. This process is described in more detail in Section 3.3.

3.2 Why a Neighbourhood Framework?

A vanilla kNN predicts a content label by aggregating the labels of k similar training instances. To this end, it uses the content as a query to retrieve neighbors from the training instances. We hypothesize that this retrieval step can be performed in a cross-lingual transfer learning scenario. In our setting, the queries are target dataset instances, and we index the source dataset for retrieval.

Note that the target instances could also be considered as neighbors for retrieval, but we exclude them, as the target dataset is small.

For a vanilla kNN model, the queries and the documents are represented using lexical features, and thus the model suffers from the curse of dimensionality (Radovanović et al., 2009). Moreover, the prediction pipeline becomes inefficient if the source dataset is considerably larger than the target dataset, as is our case here (Lu et al., 2012). Finally, for a vanilla kNN, there is no straight-forward way to map between different languages for cross-lingual transfer.

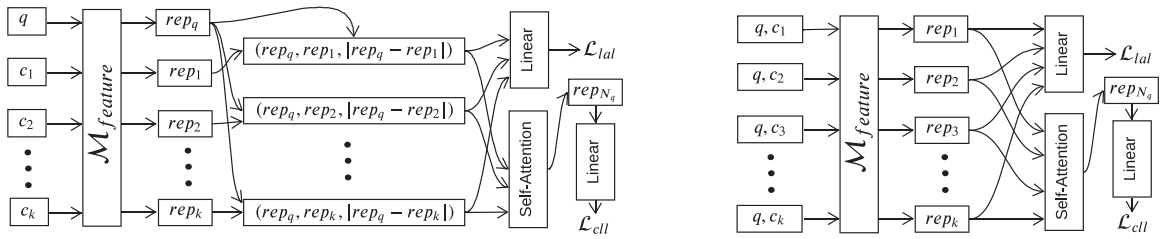
We address these problems by using a Transformer-based multilingual representation space (Feng et al., 2020) that computes the similarity between two sentences expressed in different languages. We assume that efficiency issues are less critical here for two main reasons: (*i*) retrieval using dense vector sentence embeddings has become significantly faster with recent advances (Johnson et al., 2021), and (*ii*) the number of labeled source data examples is not expected to go beyond millions, because obtaining annotations for multilingual abusive content detection is costly and the annotation process can be very harmful for the human annotators as well (Schmidt and Wiegand, 2017; Waseem, 2016; Malmasi and Zampieri, 2018; Mathur et al., 2018).

Even though multilingual language models can make the vanilla kNN model a viable solution for our problem, it is hard to make predictions with that model. Once a neighborhood is retrieved, a vanilla kNN uses a majority voting scheme for prediction, as the example in Figure 1 shows. Given a flagged Turkish query, our framework retrieves two *neutral* and one *flagged* English neighbors. Here, the majority voting prediction based on the neighborhood is incorrect. The problem is this: *A non-parametric vanilla kNN cannot make a correct prediction with an incorrectly retrieved neighborhood*. Thus, we propose a learned voting strategy to alleviate this problem.

3.3 The Architecture of kNN$^+$

We describe our kNN$^+$ framework (shown in Figure 2), including the training and the inference procedures. The framework includes neighborhood retrieval, interaction feature computation and aggregation, and a multi-task learning objective function for optimisation, which we describe in detail below.

Neighbourhood Retrieval We construct a retrieval index R from the given source dataset, $D^s = \{(x_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$. For each given example $x_i^s \in D^s$, we compute its dense vector representation, $\mathbf{x}_i^s = \mathcal{M}_{retriever}(x_i^s)$. Here, $\mathcal{M}_{retriever}$ is a multilingual sentence embedding model that we use for retrieval. There are several multilingual sentence embedding models that we could use as $\mathcal{M}_{retriever}$ (Artetxe and Schwenk, 2019; Reimers and Gurevych, 2020; Chidambaram et al., 2019; Feng et al., 2020). In this work, we



(a) Bi-Encoder kNN (BE kNN$^+$) Variant.

(b) Cross-Encoder kNN (CE kNN$^+$) Variant.

Figure 2: Two variants based on two encoding schemes used in our proposed kNN$^+$, where $\mathcal{M}_{feature}$ is the interaction feature computation model, q is the query, and c_i is a candidate neighbor. In the Bi-Encoder setup (Figure 2a), the query and each candidate are encoded separately using the same $\mathcal{M}_{feature}$ model. Afterwards, in order to obtain a joint vector representation for each query–candidate tuple, the query's representation (rep_q) is concatenated with each candidate's representation (rep_i) along with the absolute element-wise difference between the two. In the Cross-Encoder setting (Figure 2b), the query and each candidate are passed through the $\mathcal{M}_{feature}$ model, which produces the joint vector representation (rep_i) for the query–candidate tuple. Finally, we pass each joint representation through (*i*) a linear layer to predict the label agreement between the query and the candidate, and (*ii*) a self-attention layer followed by a linear projection layer to predict the label of the example.

use LaBSE (Feng et al., 2020), a strong multilingual sentence matching model, which has been trained with parallel sentence pairs from 109 languages. The model is trained on 17 billion monolingual sentences and 6 billion bilingual sentence pairs and it has achieved state-of-the-art performance for a parallel text retrieval task proposed by Zweigenbaum et al. (2017). We use $\mathbf{x}_i^s$ as a key, and we assign $(x_i^s, \mathbf{y_i}^s)$ as its corresponding value. Our retrieval index R stores all the key-value pairs computed from the source dataset.

Assume we have a training data point, $(x_j^t, y_j^t) \in D^t$, from the target dataset. We consider the content x_j^t as our query q (i.e., $q = x_j^t$). We compute a vector representation of the query, $\mathbf{q} = \mathcal{M}_{retriever}(q)$. We use $\mathbf{q}$ to score each key, $\mathbf{x}_i^s$ of R using cosine similarity (i.e., $cos(\mathbf{q}, \mathbf{x}_i^s)$).

We sort the items in R in descending order of the scores of the keys, and we take the values of the top-k items to construct the neighborhood of q, $N_q = \{(c_1, l_1), (c_2, l_2), \ldots, (c_k, l_k)\}$. Thus, each neighbor is a tuple of a content and its label from the source dataset. We convert fine-grained neighbor labels to binary labels (*flagged, neutral*) as described in Section 3.1, to align the label space with the target dataset. Nevertheless, the original fine-grained labels of the neighbors can be used to get an explanation at inference time as this is one of the core features of kNN-based models. However, our focus is on combining these models with Transformer-based ones. We leave the investigation of the explainability characteristics of kNN$^+$ for future work.

Interaction Feature Modeling As discussed in Section 3.2, the neighborhood retrieval process might lead to prediction errors. Thus, we propose a learned voting strategy to mitigate this. Our proposed strategy depends on how q relates to its neighborhood N_q. To model this relationship, we compute the interaction features between q and the content of its j-th neighbor, $c_j \in N_q$. We obtain a set of k interaction features from k neighbors, and we optimize them using query and neighbor labels.

Similarly to Reimers and Gurevych (2019), we apply two encoding schemes to compute the interaction features: A **Cross-Encoder (CE)** and **Bi-Encoder (BE)**. Under our kNN$^+$ framework, we refer to the schemes as CE kNN$^+$ for CE, and BE kNN$^+$ for BE. The BE kNN$^+$ is computationally inexpensive, while the CE kNN$^+$ is more effective. We provide a justification for this as we describe the schemes in the following paragraphs.

For the CE kNN$^+$ implementation (see Figure 2b), we first form a set of query–neighbor pairs $S_{ce} = \{(q, c_1), (q, c_2), \ldots, (q, c_k)\}$ by concatenating q with the content of each of its neighbors. Then, we obtain the output representation, $rep_j = \mathcal{M}_{feature}(q, c_j)$ of each $(q, c_j) \in S_{ce}$, from a pre-trained multilingual language model $\mathcal{M}_{feature}$. In this way, we create a set of interaction features, $I_{ce} = \{rep_1, rep_2, \ldots, rep_j\}$ from q and its neighborhood. Throughout this paper, the [CLS] token representation of $\mathcal{M}_{feature}$ is taken as its final output. We use varieties of implementations of $\mathcal{M}_{feature}$ in the experimentation.

Figure 2b shows how the interaction features are computed and optimized with a CE kNN$^+$.

Note that the feature interaction model $\mathcal{M}_{feature}$ is different from the neighborhood retrieval one $\mathcal{M}_{retriever}$. We optimize interaction features from $\mathcal{M}_{feature}$, and we leave retrieval model optimization for future work.

For the BE kNN$^+$ scheme (see Figure 2a), we obtain the output representations of q and each of the neighbors individually from $\mathcal{M}_{feature}$. Given the representation of the query, $rep_q = \mathcal{M}_{feature}(q)$, and the representation of its j^{th} neighbor, $rep_j = \mathcal{M}_{feature}(c_j)$, we model their interaction features by concatenating them along with their vector difference. The interaction features obtained for the j-th neighbor are $(rep_q, rep_j, |rep_q - rep_j|)$, and we construct a set of interaction features I_{be} from all the neighbors of q. We use the vector difference $|rep_q - rep_j|$ along with the content vectors rep_q and rep_j following the work of Reimers and Gurevych (2019). They trained a sentence embedding model using a Siamese neural network architecture with Natural Language Inference (NLI) data. They tried the following approaches to obtain features between the representations u and v of two sentences: $(u, v), (|u - v|), (u * v), (|u - v|, u * v)$, $(u, v, u * v), (u, v, |u - v|), (u, v, |u - v|), (u * v)$. Their empirical analysis showed that $(u, v, |u - v|)$ works the best for NLI data, and thus we apply this in our framework. We plan to explore other options in future work.

Both the cross-encoder and the bi-encoder architectures were shown to be effective in a wide variety of tasks including Semantic Textual Similarity and Natural Language Inference. Reimers and Gurevych (2019) showed that a bi-encoder is much more efficient than a cross-encoder, and that bi-encoder representations can be stored as sentence vectors. Thus, once $\mathcal{M}_{feature}$ is trained, the vector representations $\mathcal{M}_{feature}(x_i^s)$ of each $x_i^s \in D^s$ can be saved along with the textual contents and label. Then, at inference time, only the representation of the query needs to be computed, which reduces the computation time from $k \times \mathcal{M}_{feature}$ to a constant time. Moreover, the model can easily adapt to new neighbors without the need for retraining. However, from an effectiveness perspective, the cross-encoder is usually a better option as it encodes the query and its neighbor jointly, thus enabling multi-head attention-based interactions among the tokens of the query and of the neighbor.

Choice of $\mathcal{M}_{feature}$ We explore two $\mathcal{M}_{feature}$ models for both the CE and the BE schemes: a pre-trained XLM-R model, which we will refer to as $\mathcal{M}_{feature}^{\text{XLM-R}}$, as well as an XLM-R model augmented with *paraphrase* knowledge, which we will refer to as $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ (Reimers and Gurevych, 2020). Sentence representations from XLM-R are not aligned across languages (Ethayarajh, 2019) and $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ overcomes this problem. In particular, $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ is trained to learn sentence semantics with parallel data from 50 languages. Moreover, the training process includes knowledge distillation from a Sentence BERT model (Reimers and Gurevych, 2019) trained on 50 million English paraphrases. As such, we expect $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ to outperform $\mathcal{M}_{feature}^{\text{XLM-R}}$, as it more accurately captures the semantics of the query and its neighbor sentences. Note that there is work on producing better alignments of multilingual vector spaces (Zhao et al., 2021), which would allow us to consider a variety of pre-trained sentence representation models, but exploring this is outside the scope of this paper.

Interaction Features Optimization Given a query q and its j-th neighbor, we obtain features $rep_j \in I_{ce}$ and $(rep_q, rep_j, |rep_q - rep_j|) \in I_{be}$ from $\mathcal{M}_{feature}$ for the CE kNN$^+$ and BE kNN$^+$ schemes, respectively. For both schemes, we optimize the interaction features to indicate whether a query and its neighbor have the same or different labels. We do this to later aggregate interaction features from all the neighbors of a query to model the overall agreement of the query with the retrieved neighborhood. Our hypothesis is that understanding individual neighbor-level agreement and aggregating it will allow us also to understand the neighborhood.

We apply a fully connected layer with two outputs over the interaction features to optimize them. The outputs indicate the label agreement between q and its j-th neighbor, $(c_j, l_j) \in N_q$. There is a label agreement if both q and the j-th neighbor are flagged or are both neutral, that is, $y_j^t = l_j$. We learn the label agreement using a binary cross-entropy loss $\mathcal{L}_{lal}$, which is computed using the output of a softmax layer for each example in a batch of training data. We refer to $\mathcal{L}_{lal}$ as label-agreement loss. In our

implementation, a batch of data comprises a query and its k neighbors. We provide more details about the training procedure in Section 4.4.

Note that as our model predicts label agreement, it also indirectly predicts the label of the query and of the neighbor. In this way, it learns representations that separate flagged from the non-flagged examples.

Interaction Features Aggregation The main reasons to use interaction features for label agreement is to predict whether q should be flagged or not. In a vanilla kNN setup, there is no mechanism to back-propagate classification errors, as the only parameter to tune there is the hyper-parameter k. In our model, we propose to optimize the interaction features—using a self-attention module—to minimize the classification error with a fixed neighborhood size k. To this end, we propose to aggregate the k interaction features: I_{ce} for CE kNN$^+$ and I_{be} for BE kNN$^+$. The aggregated representation captures global information, namely, the agreement between the query and its neighborhood, whereas the interaction features capture them locally.

We use structured self-attention (Lin et al., 2017) to capture the neighborhood information. At first, we construct an interaction features matrix, $H \in \mathbb{R}^{k \times h}$ from the set of k neighbors (I_{ce} or I_{be}), where h is the dimensionality of the interaction feature space. Then, we compute structured self-attention as follows:

$$\vec{a} = \text{softmax}\left(W_2 \tanh\left(W_1 \mathbf{H}^T\right)\right) \quad (1)$$

$$\mathbf{rep_i} = \vec{a}\mathbf{H} \quad (2)$$

Here, $W_1 \in \mathbb{R}^{h_r \times h}$ is a matrix that encodes interactions between the representations and projects the interaction features into a lower-dimensional space, $h_r < h$, thus making the representation matrix $h_r \times k$ dimensional. We multiply another matrix $W_2 \in \mathbb{R}^{1 \times h_r}$ by the resulting representation, and we apply softmax to obtain a probability distribution over the k neighbors. Then, we use this probability distribution to produce an attention vector that linearly combines the interaction features to generate the neighborhood representation rep_{N_q}, which we eventually use for classification.

Classification Loss Optimization The aggregated interaction features, rep_{N_q}, are used as an input to a softmax layer with two outputs (*flagged* or *neutral*), which we optimize using a binary cross-entropy loss, $\mathcal{L}_{cll}$. We refer to $\mathcal{L}_{cll}$ as classification loss.

Optimizing this loss means that the classification decision for a query is made by computing its agreement or disagreement with the neighborhood as a whole. Our approach is a multi-task learning one, and the final loss is computed as follows:

$$\mathcal{L} = (1 - \lambda) \times \mathcal{L}_{lal} + \lambda \times \mathcal{L}_{cll} \quad (3)$$

As both the classification and the label-agreement tasks aid each other, we adopt a multi-task learning approach. We balance the two losses using the hyper-parameter λ. The classification loss forces the model to predict a label for the query. As the model learns to predict a label for a query, it becomes easier for it to reduce the label agreement loss $\mathcal{L}_{lal}$. Moreover, as the model learns to predict label agreement, it learns to compute interaction features, which represent agreement or disagreement. This, in turn, helps to optimize $\mathcal{L}_{cll}$.

Note that, at inference time, our framework requires neither the labels of the neighbors for classification, nor a heuristic-based label-aggregation scheme. The classification layer makes a prediction based on the pooled representation from the interaction features, thus removing the need for any heuristic-based voting strategy based on the labels of the neighbors. Each individual interaction feature from the query and a neighbor captures the agreement between them as we optimize the features via the L_{lal} loss. The opinion of the neighborhood is captured using an aggregation of individual interaction features—which is different from a vanilla kNN—where neighborhood opinion is captured using an individual neighbor label. As our aggregation is performed using a self-attention mechanism, we obtain a probability distribution over the interaction features that we can use to find the neighbor that influenced the neighborhood opinion the most. We also know both the original and the converted label of the neighbor (see Section 3.1 for further details about the label space conversion). The original label of the neighbor could help us understand the prediction behind the query better. For example, if the query is flagged and the original label of the most influential neighbor is *hate*, we could infer that the query is hate speech. However, we do not explore this direction in this paper, and we leave it as a future work.

4 Experimental Setting

4.1 Datasets

We conducted experiments on two different multilingual datasets covering eight languages from six language families: Slavic, Turkic, Romance, Germanic, Albanian, and Finno-Ugric. We used these datasets as our target datasets, and an English dataset as the source dataset, which contains a large number of training examples with fine-grained categorization. Both the source and target datasets are from the same domain (Wikipedia), as we do not study domain adaptation techniques in the present work. We describe these three datasets in the following paragraphs. The number of examples per dataset and the corresponding label distributions are shown in Table 1.

Jigsaw English (Jigsaw, 2018) is an English dataset with over 159K manually reviewed comments, annotated with multiple labels. We map the labels (*toxic*, *severe toxic*, *obscene*, *threat*, *insult*, and *identity hate*) into a *flagged* label; if at least one of these six labels is present for some example, we consider that example as *flagged*, and as *neutral* otherwise. As Jigsaw English is a resource-rich dataset, covering different aspects of abusive language, we use it as the source dataset. We use all its examples for training, as we validate our models on the *target* datasets' dev sets.

Jigsaw Multilingual (Jigsaw Multilingual, 2020) aims to improve toxicity detection by addressing the shortcomings of the monolingual setup. The dataset contains examples in Italian, Turkish, and Spanish. It has binary labels (toxic or non-toxic), and thus it aligns well with our experimental setup. The label distribution is fairly similar to that for Jigsaw English, as shown in Table 1. This dataset is used for experimenting in a resource-rich environment. As it does not have standard training, testing, and development sets, we split the examples in each language as follows: 1,500, 500, and 500 for Italian and Spanish, and 1,800, 600, and 600 for Turkish.

WUL (Glavaš et al., 2020) aims to create a fair evaluation setup for abusive language detection in multiple languages. Although originally in English, multilinguality is achieved by translating the comments as accurately as possible into five different languages: German (DE), Hungarian (HR), Albanian (SQ), Turkish (TR), and Russian (RU).

Dataset	Examples	Flagged %	Neutral %
Jigsaw En	159,571	10.2	89.8
Jigsaw Multi	8,000	15.0	85.0
WUL	600	50.3	49.7

Table 1: Statistics about the dataset sizes and the respective label distributions.

We use this dataset partially, by using the test set originally generated by Wulczyn et al. (2017), who focused on identifying personal attacks. In contrast to Jigsaw Multilingual, this dataset is used for experimenting in a low-resource environment. For each language, we have 600 examples, which are split as follows: 400 for training, 100 for development, and 100 for testing. As abusive content can be very culture-specific, there will be cases, even within the same language, where some utterances will be offensive in one culture, but not in another. Thus, a translation-based dataset such as WUL might not be an ideal choice, and we acknowledge this limitation.

The results from experimenting with the above datasets cannot be compared to those in the literature as we use the test set from these datasets to create our train/dev/test splits. The datasets used in previous work (Jigsaw Multilingual and WUL) provide English-only training data and observe the performance of different models in zero-shot transfer learning settings. Our setup is different as we assume that there is a limited number of training examples in the target language. Thus, we produce results only on a subset of the original testset for both datasets. Therefore, our results are not directly comparable to the results from the literature, as both the training and the testing datasets differ.

4.2 Baselines

We compare our proposed approach against three families of strong baselines. The first one considers training models only on the target dataset, the second one is source adaptation, where we use Jigsaw English as our source dataset, and the third one consists of traditional kNN classification method, but with dense vector retrieval using LaBSE (Feng et al., 2020). We use cosine similarity under a LaBSE representation space to

retrieve neighbors for the baselines and for our proposed approaches.

Target Dataset Training This family of baselines uses only the target dataset for training:

Lexicon approach: After standard text tokenization and normalization of the text, we count the number of terms it contains that are also listed in the abusive language lexicon HurtLex.[3] Based on the development set, we learn a threshold for the minimum number of matches required to flag the text. Then, we apply the lexicon and the threshold to the test set.

fastText is a baseline that uses the mean of the token vectors obtained from fastText (Joulin et al., 2017) word embeddings to represent a textual example. These representations are then used in a binary logistic regression classifier.

XLM-R Target is a pre-trained XLM-R model, which we fine-tune on the target dataset.

Source Adaptation This family of baselines includes variations of XLM-R:

XLM-R Mix-Adapt is a baseline model, which we train by mixing source and target data. This is possible because the label inventories of our source and target datasets are the same: $\mathcal{Y} = \{flagged, neutral\}$. The mixing is done by oversampling the target data to match the number of instances of the source dataset. As the number of instances in the target dataset is limited, this is preferable to undersampling.

XLM-R Seq-Adapt (Garg et al., 2020) is a Transformer pre-trained on the source and fine-tuned on the target data. Here, we fine-tune XLM-R on the Jigsaw English dataset, and then we do a second round of fine-tuning on the target dataset.

Nearest Neighbor We apply two nearest neighbor baselines, using majority voting for label aggregation. We varied the number of neighbors from 3 to 20, and we found that using 10 neighbors works best (on the dev set).

LaBSE-kNN Here the source dataset is indexed using representations obtained from LaBSE sentence embeddings (Feng et al., 2020), and the neighbors are retrieved using cosine similarity.

Weighted LaBSE-kNN is a baseline that uses the same retrieval step as LaBSE-kNN, but with a weighted voting strategy: Each label is scored by summing the cosine similarities for the retrieved flagged and neutral neighbors, respectively; then, the label with the highest score is returned.

[3] https://github.com/valeriobasile/hurtlex.

4.3 Evaluation Measures

Following prior work on abusive language detection, we use F1 measure for evaluation. The F1 measure combines precision and recall (using a harmonic mean), which are both important to consider for automatic abusive language detection systems. In particular, online platforms strive to remove all content that violates their policies, and thus, if the system were to achieve 100% recall, the contents could be further filtered by human moderators to weed out the benign content. However, if the system's precision were very low, it would mean that the moderators would have to read every piece of content on the platform.

4.4 Fine-Tuning and Hyper-Parameters

We train all the models for 10 epochs with XLM-R as a base transformer representation with a maximum sequence length of 256 tokens. However, we make an exception for SRC (see Section 5.1): We train it for a single epoch, as training a neighborhood-based model on a large dataset is resource-intensive. For all the approaches, we use Adam with β_1 0.9, β_2 0.999, ϵ 1e-08 as the optimizer setting. For the baseline models, we use a batch size of 64, and a learning rate of 4e-05. For kNN$^+$-based models, we create a training batch from a query and its 10-nearest neighbors. For stable updates, we accumulate gradients from 50 batches before back-propagation. We selected the values of all of the aforementioned hyper-parameters based on the validation set. For kNN$^+$-based models, the best learning rate is selected from {5e-05, 7e-05}.

5 Experimental Results

5.1 Evaluation in a Cross-lingual Setting

Table 2 shows the performance of our model variants compared to the seven strong baselines we described above (rows 1–7). The first two rows represent non-contextual baselines and they perform worse compared to the baseline pre-trained XLM-R models fine-tuned with labeled data (rows 3–5). Specifically, the lexicon baseline performs the worst among all, which indicates the limited coverage of hate speech lexicon and the loss in precision due to token mismatches and context

#	Method	Jigsaw Multilingual			WUL					
		ES	IT	TR	DE	EN	HR	RU	SQ	TR
1	Lexicon	35.8	40.5	34.0	70.9	70.6	63.9	63.6	58.2	71.8
2	FastText	55.3	47.2	64.2	74.2	72.7	58.9	74.2	65.9	72.5
3	XLM-R Target	63.5	56.4	80.6	82.1	75.7	73.2	76.7	77.3	78.8
4	XLM-R Mix-Adapt	**64.2**	58.5	76.1	83.2	**93.9**	87.3	82.1	86.2	86.0
5	XLM-R Seq-Adapt	60.5	58.3	81.2	83.9	88.0	80.0	80.0	86.3	83.5
6	LaBSE-kNN	44.7	48.5	66.0	70.8	77.1	84.1	79.1	83.1	75.6
7	Weighted LaBSE-kNN	44.8	38.3	52.1	71.7	85.4	82.4	79.5	83.7	81.0
8	CE kNN$^+$ + $\mathcal{M}^{\text{XLM-R}}_{feature}$	58.9	63.8	78.5	80.4	83.8	86.2	77.6	83.5	85.4
9	CE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature}$	59.4	**67.0**	84.4	84.8	88.0	86.3	83.8	83.0	86.5
10	CE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ $\rightarrow$ SRC	61.2	61.1	**85.0**	**89.5**	92.3	**90.6**	84.9	89.5	87.3
11	BE kNN$^+$ + $\mathcal{M}^{\text{XLM-R}}_{feature}$	52.2	60.3	75.0	81.6	80.8	77.9	78.0	79.6	79.6
12	BE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature}$	58.8	56.6	80.6	83.8	86.9	82.2	**86.9**	84.9	83.7
13	BE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ $\rightarrow$ SRC	59.1	59.5	81.6	88.7	90.7	87.6	86.3	**90.2**	**88.7**

Table 2: Comparison of F1 scores for the baselines and for our model variants. BE kNN$^+$ and CE kNN$^+$ indicate Bi-encoder and Cross-encoder schemes, respectively. SRC indicates that the model was further pre-trained with source Jigsaw English, using data from it as both query and neighbors.

obliviousness. For example, the word *monkey* is generally included in a hate speech lexicon, but the appearance of the token in a textual content does not necessarily mean that the content is abusive.

The highlighted rows in Table 2 show different variants of our framework, based on CE kNN$^+$ and BE kNN$^+$, that is, using cross-encoders vs. bi-encoders. For each of the encoding schemes, we instantiate three different models by using three different pre-trained representations fine-tuned in our neighborhood framework, namely: $\mathcal{M}^{\text{XLM-R}}_{feature}$, which is a pre-trained XLM-RoBERTa model (XLM-R); $\mathcal{M}^{\text{P-XLM-R}}_{feature}$, which is an XLM-R model fine-tuned under a knowledge distillation setting with 50 million paraphrases and parallel data in 50 languages (Reimers and Gurevych, 2020); and $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ $\rightarrow$ SRC, which is an $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ model fine-tuned with source data (here, 159,571 instances from Jigsaw English) in our neighborhood framework.

In order to train with SRC, we use all the training data in Jigsaw English, and we retrieve neighbors from Jigsaw English using LaBSE sentence embeddings.[4] Then, we use this training data to fine-tune $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ with our kNN$^+$-based cross-encoder (CE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ $\rightarrow$ SRC) and bi-encoder (BE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ $\rightarrow$ SRC) experimental setups.

This is analogous to applying sequential adaptation (Garg et al., 2020), but here we do it in our neighborhood framework.

The SRC approach addresses one of the weaknesses of our kNN framework. The training data is created from instances in the target dataset and their neighbors from the source dataset. Thus, the neighborhood model cannot use all source training data, as it pre-selects a subset of the source data based on similarity. This is a disadvantage compared to the sequential adaptation model, which uses all source training instances for pre-training. In order to overcome this, we use the neighborhood approach to pre-train our models with source data.

Table 2 shows the F1 scores for eight language-specific training and evaluation sets stemming from two different data sets: Jigsaw Multilingual and WUL. Jigsaw Multilingual is an imbalanced dataset with 15% abusive content and WUL is balanced (see Table 1). Thus, it is hard to achieve high F1 score in Jigsaw Multilingual,

[4]Note that we only use LaBSE for retrieval, as it has a large coverage of languages.

whereas for WUL the F1 scores are relatively higher. Our CE kNN$^+$ variants achieve superior performance to all the baselines and our BE kNN$^+$ variants as well in the majority of the cases.

The performance of the best and of the second-best models for each language are highlighted by **bold-facing** and underlining, respectively. We attribute the higher scores achieved by CE kNN$^+$ variants compared to the BE kNN$^+$ on the late-stage interaction of the query and its neighbors.

The CE kNN$^+$ variants show a large performance gain compared to baseline models on the Italian and the Turkish test sets from Jigsaw Multilingual. Even though the additional SRC pre-training is not always helpful for the CE kNN$^+$ model, it is always helpful for the BE kNN$^+$ model. However, both models struggle to outperform the baseline for the Spanish test set. We analyzed the training data distribution for Spanish, but we could not find any noticeable patterns.

Yet, it can be observed that the XLM-R target baseline for Spanish (2nd row, 1st column) achieves a higher F1 score compared to the Seq-Adapt baseline, which yields better performance for Italian and Turkish. We believe that the in-domain training examples are good enough to achieve a reasonable performance for Spanish.

On the WUL dataset, BE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ with SRC pre-training outperforms the CE kNN$^+$ variants and all baselines for Albanian, Russian, and Turkish. Both the BE kNN$^+$ variants and the CE kNN$^+$ variants perform worse compared to the XLM-R Mix-Adapt baseline for English. Seq-Adapt is a recently published effective baseline (Garg et al., 2020), but for the WUL dataset, it does not perform well compared to the Mix-Adapt baseline. Note that the test set for the WUL dataset is relatively small (100 examples per language) and the examples in the test set are human translations of the English test set. Yet, we chose this dataset as it results in a larger coverage of languages. We acknowledge this limitation (that the dataset is based on translations) in our experiments and that is why we further use Jigsaw Multilingual to demonstrate the generality of our results.

5.2 Impact of the Learned Voting Strategy

To demonstrate the effectiveness of our learned voting strategy, we use our baselines (shown in Table 2, rows 3–7) to retrieve neighbors, and then we perform majority voting to predict the label of a test instance. The results for all the approaches are shown in Table 3. For comparison, we also add the best bi-encoder and cross-encoder versions of kNN$^+$ (see Table 2, rows 10 and 13).

Method	**ES**	**IT**	**TR**
Fine-Tuned kNN Baselines			
XLM-R Target-kNN	32.3	23.8	48.5
XLM-R Mix-Adapt-kNN	40.9	30.3	38.2
XLM-R Seq-Adapt-kNN	29.7	34.9	32.2
Sentence Similarity kNN Baselines			
LaBSE-kNN	44.7	48.5	66.0
Weighted LaBSE-kNN	44.8	38.3	52.1
Our Model			
BE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}} \rightarrow$ SRC	59.1	59.5	81.6
CE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}} \rightarrow$ SRC	**61.2**	**61.1**	**85.0**

Table 3: Performance comparison in terms of F1 score for the baseline classification models and the sentence similarity model LaBSE under the majority voting kNN setup (experiments on Jigsaw Multilingual).

In particular, these baseline models are pre-trained XLM-R models fine-tuned on different combinations of source and target language datasets (see *Fine-Tuned kNN Baselines*, Table 3). For each data case in the source dataset, we compute its representation as the [CLS] token from the classification model and we construct a list of vectors. Given a test data case from the target dataset, we also compute its representation based on the [CLS] token representation from the classification model. We then compute its cosine similarity with each of the [CLS] vectors from the source dataset. After that, we compute a ranked list of the top-10 neighbors based on similarity scores.

Next, we vary the number of neighbors from three to ten—considering them in the order they are ranked based on their similarity to the query—to obtain a majority vote and to classify the test example. We can see in Table 3 that the performance is similar to that for the LaBSE-kNN and for the Weighted LaBSE-kNN approaches in which the neighbors are retrieved using a representation space constructed from sentence similarity data (see *Sentence Similarity kNN Baselines*, Table 3). The results in Table 3 show that when fine-tuned models are directly used in a nearest neighbors framework without additional modifications, their performance is lower by between

25 and 60 F1 points absolute, compared to our proposed kNN$^+$ model.

These results suggest that the interactions between the query and the retrieved neighbors captured by our model are an important prerequisite for achieving high performance.

5.3 Evaluation in a Multilingual Setting

In this subsection, we go beyond our cross-lingual setting and we analyse the effectiveness of our proposed model in a multilingual setting. A multilingual setting has been explored in recent work on abusive language detection (Pamungkas and Patti, 2019; Ousidhoum et al., 2019; Basile et al., 2019; Ranasinghe and Zampieri, 2020; Corazza et al., 2020; Glavaš et al., 2020; Leite et al., 2020 and it is desirable because online platforms are not limited to specific languages. An effective multilingual model unifies the two-stage process of language detection and prediction with a language-specific classifier. Moreover, abusive language is generally code-mixed (Saumya et al., 2021), which makes language-agnostic representation spaces more desirable.

We investigate a multilingual scenario, where all target languages in our cross-lingual setting are observed both at training and at testing time. To this end, we create new training, development, and testing splits in a 5:1:2 ratio from the 8,000 available data cases in the Jigsaw Multilingual dataset. Each split contains randomly sampled data in Italian, Spanish, and Turkish.

We train and evaluate our BE kNN$^+$ and CE kNN$^+$ using the aforementioned splits; the results are shown in Table 4. Here, we must note that our neighborhood retrieval model is language-agnostic, and thus we can retrieve neighbors for queries in any language.

We find that in a multilingual scenario, our BE kNN$^+$ model with SRC pre-training performs better than the CE kNN$^+$ model. Both the BE and the CE approaches supersede the best baseline model Seq-Adapt. Compared to the cross-lingual setting, there is more data in a mix of languages available. We hypothesize that the success of the bi-encoder model over the cross-encoder one stems from the increase in data size.

5.4 Analysis of the BE Representation

In order to understand the impact of the representations by BE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature} \rightarrow$ SRC, a model variant instantiated from our proposed kNN framework, we computed the similarity between the query and its neighbors in the representation space. An example is shown in Table 5 (it is the example from the Introduction). Given the Turkish flagged query, we use LaBSE (Feng et al., 2020) and our BE representation space to retrieve ranked lists of its ten nearest neighbors. The table shows the scores computed by both approaches, and we can see that our representation can help discriminate between flagged and neutral contents better. When we compute the cosine similarity between the query and the nearest neighbors, the BE representation space assigns negative scores to the neutral content. The LaBSE sentence embeddings are optimized for semantic similarity, and thus using them does not allow us to discriminate between flagged and neutral content.

Model	**Representations**	**F1**
Seq-Adapt	XLM-R	64.4
CE-kNN	$\mathcal{M}^{\text{XLM-R}}_{feature}$	64.2
	$\mathcal{M}^{\text{P-XLM-R}}_{feature}$	62.8
	$\mathcal{M}^{\text{P-XLM-R}}_{feature} \rightarrow$ SRC	65.1
BE-kNN	$\mathcal{M}^{\text{XLM-R}}_{feature}$	65.5
	$\mathcal{M}^{\text{P-XLM-R}}_{feature}$	63.7
	$\mathcal{M}^{\text{P-XLM-R}}_{feature} \rightarrow$ SRC	**67.6**

Table 4: Effectiveness of our BE kNN$^+$ and CE kNN$^+$ schemes in the multilingual setting that we create from Jigsaw Multilingual.

Text	**BE kNN$^+$** Score	**LaBSE** Score	**Label**
[redacted] off i do what i want	0.99	0.88	flagged
[redacted] you i do what i want	1.0	0.84	flagged
i have going to do what ever i want	−0.19	0.83	neutral
[redacted] u i will do as i please	1.0	0.81	flagged
[redacted] off [redacted] off i do what i want [redacted]	1.0	0.77	flagged
nah man i do wat i want	−0.19	0.75	neutral
i shall go ahead and do it	−0.18	0.74	neutral
whaaat whateva i do what i wantâĂİ	−0.2	0.72	neutral
ok i will do it	−0.17	0.69	neutral
great i will do what you are saying	−0.16	0.68	neutral

Table 5: An example showing the effectiveness of our bi-encoder representation space for computing the similarity between the query (flagged) and its neighbors. We masked the offensive tokens in the examples for better reading experience.

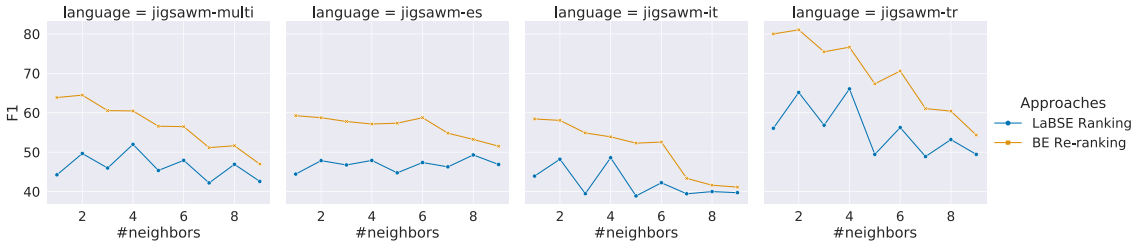


Figure 3: Impact of re-ranking neighbors using LaBSE in the BE kNN$^+$ representation space.

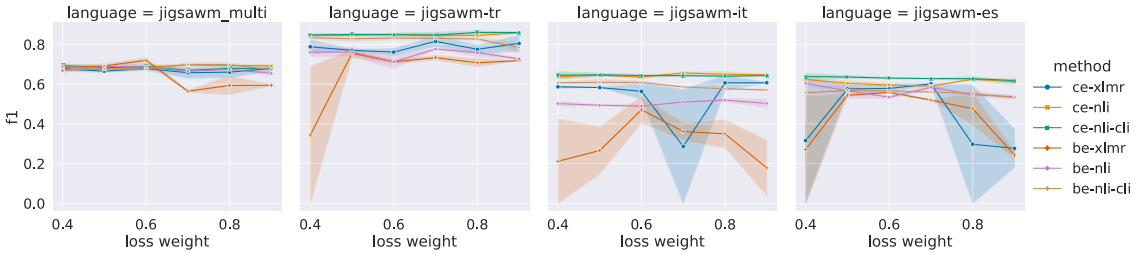


Figure 4: Multi-task loss parameter sensitivity with uncertainties from two learning rates: 5e-05, 7e-05.

We further study the impact of our representation by comparing a voting-based kNN on the top-10 neighbors retrieved by LaBSE vs. a re-ranking using our BE representation.

For both the LaBSE-based ranking and for our re-ranking, at each ranking point, we apply the majority voting kNN approach on the neighborhood within that ranking point. Figure 3 shows the results for the test part of the Jigsaw Multilingual dataset (including the multilingual setup; see Section 5.3). We can see that the re-ranking step improves over LaBSE for all the different numbers of neighbors.

5.5 Multi-Task Learning Parameter Sensitivity

Our approach uses multi-task learning, where we balance the weights of $\mathcal{L}_{cll}$ and $\mathcal{L}_{lal}$ using a hyper-parameter λ. Figure 4 shows the impact of different values for this hyper-parameter. On the horizontal axis, we increase the importance of the $\mathcal{L}_{lal}$ loss, and we show the performance of all model variants on the development part of the Jigsaw Multilingual dataset. We can see that the models perform well if the weight for the label-agreement loss is set to 0.7, and degrades if it is increased.

6 Conclusion and Future Work

We proposed kNN$^+$, a novel framework for cross-lingual content flagging, which significantly outperforms strong baselines with limited training data in the target language. We further demonstrated the effectiveness of our framework in a multilingual scenario, where a test data point can be in Turkish, Italian, or Spanish.

Moreover, we provided a qualitative analysis of the representations learned by our proposed BE kNN$^+$ framework, and we demonstrated that, in the learned representation space, flagged content stays close to flagged content, while non-flagged stays close to non-flagged content.

Our framework computes a neighborhood representation for a query using an attention mechanism, thus indicating the influence of each individual neighbor. This and the kNN-based architecture offer an opportunity to obtain an explanation for the individual model predictions, and such explanations can be based not only on the textual content of the influential neighbors, but also on their original fine-grained labels.

In future work, we plan to understand the viability of such explanations in a user study. We also plan to evaluate our framework on other content flagging tasks, e.g., for detecting harmful memes (Dimitrov et al., 2021; Pramanick et al., 2021a,b), as the framework is not limited to abusive content detection.

Acknowledgments

We would like to thank the entire Checkstep team for the useful discussions on the potential

implications of this research. We would especially like to thank Jay Alammar, who further provided feedback on the model and created the general conceptual diagram that explains our proposed neighborhood framework.

References

Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *Proceedings of the 40th European Conference on IR Research (ECIR 2018)*, volume 10772 of *Lecture Notes in Computer Science*, pages 141–153. Springer. https://doi.org/10.1007/978-3-319-76941-7_11

Aymé Arango, Jorge Pérez, and Barbara Poblete. 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*, SIGIR'19, pages 45–54, Paris, France. Association for Computing Machinery. https://doi.org/10.1145/3331184.3331262

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610. https://doi.org/10.1162/tacl_a_00288

Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *Proceedings of the World Wide Web Conference*, WWW '19, pages 49–59, San Francisco, CA, USA. Association for Computing Machinery. https://doi.org/10.1145/3308558.3313504

Michele Banko, Brendon MacKeen, and Laurie Ray. 2020. A unified taxonomy of harmful content. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 125–137, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.alw-1.16

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. https://doi.org/10.18653/v1/S19-2007

Antal van den Bosch, Bertjan Busser, Sander Canisius, and Walter Daelemans. 2007. An efficient memory-based morphosyntactic tagger and parser for dutch. *LOT Occasional Series*, 7:191–206.

Muthu Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yunhsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Learning cross-lingual sentence representations via a multi-task dual-encoder model. In *Proceedings of the 4th Workshop on Representation Learning for NLP*, RepL4NLP '19, pages 250–259, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-4330

European Commission. 2020. Shaping Europe's digital future: The digital services act package. https://doi.org/10.18653/v1/2020.acl-main.747

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32, Vancouver, Canada. Curran Associates, Inc.

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020. A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology*, 20(2). https://doi.org/10.1145/3377323

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In *Proceedings of the Fourth Workshop on Very Large Corpora*, Herstmonceux Castle, Sussex, UK. Association for Computational Linguistics.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, ALW '19, pages 25–35, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-3504

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186. Minneapolis, Minnesota, USA.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting propaganda techniques in memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL '21, pages 6603–6617, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.516

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP '19, pages 55–65, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1006

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2021. Augmenting transformers with knn-based composite memory for dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99. https://doi.org/10.1162/tacl_a_00356

Elise Fehn Unsvåg and Björn Gambäck. 2018. The effects of user features on Twitter hate speech detection. In *Proceedings of the 2nd Workshop on Abusive Language Online*, ALW '18, pages 75–85, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5110

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *arXiv preprint arXiv:2007.01852*.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7780–7788. https://doi.org/10.1609/aaai.v34i05.6282

Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos. 2018. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, SETN '18, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3200947.3208069

Goran Glavaš, Mladen Karan, and Ivan Vulić. 2020. XHate-999: Analyzing and detecting abusive language across domains and languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, COLING '20, pages 6350–6365, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.559

UK Government. 2020. Online harms white paper.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM'16, pages 55–64, New York, NY, USA. Association for Computing Machinery.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938, Online. PMLR.

Jigsaw. 2018. Toxic comment classification challenge. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/. Online; accessed 28 February 2021.

Jigsaw Multilingual. 2020. Jigsaw multilingual toxic comment classification. https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification/. Online; accessed 28 February 2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547. https://doi.org/10.1109/TBDATA.2019.2921572

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '17, pages 427–431, Valencia, Spain. Association for Computational Linguistics. https://doi.org/10.18653/v1/E17-2068

David Jürgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 3658–3666, Florence, Italy. Association for Computational Linguistics.

Lukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. 2017. Learning to remember rare events. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, Toulon, France. OpenReview.net.

Nora Kassner and Hinrich Schütze. 2020. BERT-kNN: Adding a kNN search component to pretrained language models for better QA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, EMNLP '20, pages 3424–3430, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.307

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *Proceedings of the 8th International Conference on Learning Representations*, ICLR '20, Addis Ababa, Ethiopia. OpenReview.net.

João Augusto Leite, Diego Silva, Kalina Bontcheva, and Carolina Scarton. 2020. Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, AACL '20, pages 914–924, Suzhou, China. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33 of *NeurIPS '20*, pages 9459–9474. Curran Associates, Inc.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *The 5th International Conference on Learning Representations, ICLR '17*. Toulon, France.

Wei Lu, Yanyan Shen, Su Chen, and Beng Chin Ooi. 2012. Efficient processing of k nearest neighbor joins using mapreduce. *Proceedings of the VLDB Endowment*, 5(10):1016–1027. https://doi.org/10.14778/2336664.2336674

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS One*, 14(8).

https://doi.org/10.1371/journal.pone.0221152, PubMed: 31430308

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202. https://doi.org/10.1080/0952813X.2017.1409284

Puneet Mathur, Rajiv Ratn Shah, Ramit Sawhney, and Debanjan Mahata. 2018. Detecting offensive tweets in Hindi-English code-switched language. In *SocialNLP@ACL*, pages 18–26. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-3504

Preslav Nakov, Vibha Nayak, Kyle Dent, Ameya Bhatawdekar, Sheikh Muhammad Sarwar, Momchil Hardalov, Yoan Dinkov, Dimitrina Zlatkova, Guillaume Bouchard, and Isabelle Augenstein. 2021. Detecting abusive language on online platforms: A critical analysis. *arXiv preprint arXiv:2103.00153*.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4675–4684, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1474

Endang Wahyu Pamungkas and Viviana Patti. 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*, pages 363–370, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-2051

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1117

Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021a. Detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2783–2796, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.246

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021b. MOMENTA: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455, Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.379

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2009. Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 865–872, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/1553374.1553485

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5838–5844, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.470

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*,

pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics. https://doi.org.org/10.18653/v1/2020.emnlp-main.365

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2021. SOLID: A large-scale semi-supervised dataset for offensive language identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 915–928, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.80

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh. 2021. Offensive language identification in dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 36–45, Kyiv, Ukrane. Association for Computational Linguistics.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *SocialNLP@EACL*, pages 1–10. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-1101

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it blend? Blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL '18, pages 599–605, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2095

Saurabh Srivastava, Prerna Khurana, and Vartika Tewari. 2018. Identifying aggression and toxicity in comments using capsule network. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying*, TRAC '18, pages 98–105, Santa Fe, New Mexico, USA. Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-3517

Lukas Stappen, Fabian Brunn, and Björn Schuller. 2020. Cross-lingual zero-and few-shot hate speech detection utilising frozen transformer language models and AXEL. *arXiv preprint arXiv:2004.13850*.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, ALW '19, pages 80–93, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-3509

Eric Wallace, Shi Feng, and Jordan Boyd-Graber. 2018. Interpreting neural networks with nearest neighbors. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 136–144, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5416

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

Zeerak Waseem. 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*, pages 138–142, Austin, Texas. Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-5618

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at

scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1391–1399, Geneva, Switzerland. International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3038912.3052591

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 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*, NAACL-HLT '19, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1144

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. 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. https://doi.org/10.18653/v1/S19-2010g

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. International Committee for Computational Linguistics. https://doi.org/10.18653/v1/2020.semeval-1.188

Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2021. Inducing language-agnostic multilingual representations. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 229–240, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.starsem-1.22

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, Vancouver, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-2512