electronics

MDPI

Article

Development of an Automated Moderator for Deliberative Events

Simone Bonechi

Department of Social, Political and Cognitive Science, University of Siena, 53100 Siena, Italy; simone.bonechi@unisi.it

Abstract: Online communication platforms have revolutionized interpersonal interactions by transcending geographical barriers. While facilitating connectivity, these platforms have introduced challenges such as overcoming linguistic differences and preventing spam and offensive content diffusion. This is particularly pertinent in the context of deliberative events, where online platforms could be used to extend the inclusion of citizens in democratic decision-making. In traditional deliberative events, human moderators and translators were used to facilitate conversation; however, the need for these figures imposed a limit on both the number of deliberative events that could be organized and the number of participants. In response, this paper proposes an automated moderator for deliberative events. The moderator is developed in Python for the online communication platform Discord and can be used, thanks to the integrated AI (Artificial Intelligence) tools, to automatically manage conversation agendas, prevent spam and inappropriate language, analyze the sentiment of the conversation, and translate messages into multiple languages. In particular, three classifiers, based on a pre-trained BERT (Bidirection Encoder Representations from Transformers), were fine-tuned for spam detection, toxic comments classification, and sentiment analysis. These allow the moderator to automatically detect and remove spam and offensive messages in different languages, send warnings to users, alert administrators, and, after repeated warnings, impose bans. Additionally, a built-in translator, based on Meta's No Language Left Behind NLLB model, translates messages into five languages (Italian, English, French, German, and Polish). The developed bot was tested in a simulated deliberative event on a Discord server, demonstrating its ability to manage conversations and prevent linguistic abuse.

Keywords: automated moderation; text classification; spam classification; toxic comment detection

Citation: Bonechi, S. Development of an Automated Moderator for Deliberative Events. *Electronics* **2024**, *13*, 544. https://doi.org/10.3390/electronics13030544

Received: 27 December 2023
Revised: 25 January 2024
Accepted: 27 January 2024
Published: 29 January 2024

1. Introduction

In recent years, especially after the coronavirus outbreak, online communication platforms have significantly changed the way people interact. Thanks to these platforms, it is possible to overcome geographical distance by allowing different communities to connect and exchange ideas. On the one hand, this eliminated the need for physical proximity and made communication easier, but, on the other, it brought about some major challenges. Issues such as spam, offensive content, and language diversity barriers have become problems that need to be addressed if we want to create a secure environment for virtual interactions. This is especially true in the context of deliberative events [1], where participants engage in thoughtful and collaborative discussions to make informed decisions on a particular topic. Such events often involve participants with diverse perspectives to foster the exchange of ideas and encourage democratic decision-making. Deliberative events are characterized by an emphasis on informed dialogue and active listening to explore diverse perspectives and reach thoughtful conclusions. These events are conducted in both physical and online settings, typically featuring a moderator to facilitate discussions, discourage the use of offensive language, and provide translations when participants speak different languages. With these kinds of discussions, it is possible to promote citizen

involvement in governance, offering a platform to rebuild trust in government institutions, empower authorities, and solicit high-quality input for effective governance. While virtual environments represent an opportunity to expand the frequency of these events, allowing for broader participant inclusion in governance, the need for human moderators (HMs) and translators imposes significant constraints on scalability. Removing the moderator could drastically impact the quality of discourse, potentially leading to disrespectful dialogue and decreasing the effectiveness of deliberations. At the same time, the absence of a translator creates a problem in terms of participation, limiting accessibility to international audiences.

For the above reasons, in this paper, I propose an automated moderator (AM) that can be employed in a deliberative event to automatically manage the agenda of the conversation, prevent spam and inappropriate language, and translate the messages of the participants into five different languages (Italian, English, French, German, and Polish). We studied a number of platforms, including Nextcloud, Decidim, Jitsi, Discord, and other open-source software, to identify the one that has the most suitable features for AM development. At the end of this preliminary study, we decided to employ Discord, a popular communication platform designed for creating communities, connecting people, and facilitating real-time communication. Discord was initially developed for gamers, but its versatile features have led to its widespread use across various communities. It is organized into servers, which are like communities that can be created by users. Within each server, users can create different channels, which can be text and/or voice, for specific topics or activities. Furthermore, it allows various forms of multimedia sharing, including images, videos, and links, enriching the communication experience. A hierarchical user organization in the server is possible due to the ability to assign roles and permissions to users; there can be, for instance, roles such as "Administrator", "Moderator", and "Member", each with different levels of access and control. The main reasons that led to the choice of Discord are as follows:

- Security and Privacy: Discord places a strong emphasis on user privacy and security. Conversations are encrypted, and the platform provides features like two-factor authentication to enhance account security;
- Cross-Platform Availability: Discord is available on multiple platforms, including Windows, macOS, Linux, iOS, and Android. This makes it accessible to a wide range of users, regardless of their preferred device;
- Bots: Discord allows the integration of bots, which are automated programs that can perform various tasks.

Choosing Discord as a communication platform allows us to have stable servers and to focus the study only on the development of the AM and not on the platform. Furthermore, thanks to discord.py (https://discordpy.readthedocs.io/en/stable/, accessed on 12 December 2023), a Python library that allows one to create bots and easily integrate them into a Discord server, it is also possible to extend the bot's ability with artificial intelligence (AI) features developed using PyTorch and/or TensorFlow.

Indeed, in recent years, AI—especially thanks to deep learning (DL)—has achieved notable successes in several fields, including computer vision [2,3], biomedicine [4–7], and natural language processing [8,9]. In particular, for natural language understanding, Transformer-based encoder models, like BERT [10], dramatically outperform previous state-of-the-art methods.

Following these successes, in this work, we decided to exploit DL to develop an AM for integration into a Discord server. The moderator is able to welcome the users, send them all the information and material needed in the discussion, manage the roles, and prevent abuse during the conversation. Three classifiers were trained to automatically detect spam, classify offensive messages, and to perform sentiment analysis. All the classifiers were based on a pre-trained BERT model; the spam detector was fine-tuned on UtkMl's Twitter Spam Detection dataset (https://www.kaggle.com/c/utkmls-twitter-spam-detection-competition/overview, accessed on 15 June 2023), while the Jigsaw Multilingual Toxic Comment dataset (https://www.kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification/overview, accessed on 15 June 2023) was used

to fine-tune the offensive message classifier; finally, the Large Movie Review Dataset (https://ai.stanford.edu/~amaas/data/sentiment/, accessed on 15 June 2023) [11] was used for sentiment analysis. Thanks to these AI tools, the moderator can automatically remove inappropriate messages, give a warning to the user who sends the message, and send an alert to the administrator of the server. After a given number of warnings, the moderator automatically bans the user from the server.

Moreover, we have integrated a translator into the moderator based on the NLLB (No Language Left Behind) model [12] proposed by Meta, to automatically translate messages into five distinct languages. In the current implementation of the moderator, we have used only five languages (Italian, English, French, German, and Polish) to reduce the required computational load. Nevertheless, given that, for the NLLB project, META trained models for an extensive array of languages, there is potential to expand the translation capabilities to include additional languages. The combination of the translator with the other AI tools allows the bot to prevent the use of inappropriate content in all five languages.

The developed AM operates in near real-time; this is a key feature that is necessary to not impact the effectiveness of the deliberative event. For this reason, the selection of artificial intelligence tools included in AM was made carefully, trying to balance accuracy and computational efficiency. Indeed, the model chosen for toxic detection, spam detection, and sentiment analysis is capable of analyzing a sentence in less than a second (approximately 0.044 s) and the translator translates a sentence into five languages in about two seconds.

The developed bot was included in a Discord server and tested in a simulated deliberative event, demonstrating its potential and usefulness to manage the conversation and prevent linguistic abuse.

The main contributions of the paper can be summarized as follows:

- This is the first AM specifically designed for deliberative events;
- The proposed AM is the first to integrate a translator to effectively facilitate conversation between five different languages;
- AM integrates three AI tools specifically trained for toxic comment detection, spam detection, and sentiment analysis.

The paper is organized as follows. In Section 2, the literature related to the main aspects connected to the AM is revised. Section 3 describes the features of the developed bot, and Section 4 presents the training and the evaluation of the AI tools integrated into the bot. Finally, Section 5 draws conclusions and discusses possible future developments.

2. Related Works

For decades, researchers have explored the role of online moderators from a socio-technological [13] and legal [14] perspective. A key aspect of moderation revolves around overseeing the content of conversations [15]. In this context, both automated and human moderators play a crucial role by reporting content that violates platform policies. Convicted users may be subject to sanctions such as removal of content [16] and account suspension [17]. Human moderators evaluate content based on their expertise and experience, while AMs rely on tools that can range from regular expressions [18] to artificial intelligence for decision-making.

2.1. Toxic Comment Detection

One of the first AI tools involved in moderation was proposed by [19], where a decision tree trained on predefined rules was employed to classify offensive messages. With a similar goal, in [20], the authors proposed a multi-step approach that combines various classifiers with an underlying dictionary of offensive and abusive phrases. A bag-of-words approach was used to train a model to recognize offensive messages in [21]. While this method may not take advantage of sentence syntax and word order, it has been proven to produce highly effective results. In [22], various approaches, including logistic regression [23], Naive Bayes [24], decision trees [25], random forests [26] and Support Vector Machines (SVMs) [27], have been compared for hate speech detection. The results

indicated that SVMs and logistic regressors guarantee superior performance. The challenge of data scarcity in this domain, highlighted by [28], led to a solution proposed by [29], suggesting the use of automatically labeled data to overcome this limitation.

In recent years, instead, DL has taken the lead in the field. A deep neural network is exploited in [30] to learn low-dimensional distributed representations of comments that can be used as input to a logistic regression model that is capable of classifying hate speech. In [31–33], deep neural networks with attention layers and recurrent neural networks were used to moderate user comments. Furthermore, approaches based on LSTMs [34] and Convolutional Neural Networks (CNNs) [35] have been studied by [36–39]. However, these models often show lower accuracy than modern Transformer-based approaches [8]. Finally, more related to this work, a classifier based on the Bidirectional Encoder Representation from Transformers (BERT) has proven effective for hate speech detection, as demonstrated by [40–42].

2.2. Spam Detection

Over the years, a myriad of machine learning approaches have been designed to detect spam in emails and messages. In [43], K-Nearest Neighbor, Naive Bayes, and inverse DBSCAN algorithms were combined to develop a spam email detector that was capable of analyzing both text and images. Instead, a Word Sense Disambiguation preprocessing was used to prepare the input data for a machine learning model in [44]. In [45], the authors proposed a combination of TF–IDF (Term Frequency–Inverse Document Frequency) and SVMs for effective spam detection.

More recently, DL has also been leveraged for spam detection. A modified Transformer was employed for this purpose in [46], while a fine-tuned version of BERT was applied in [47,48].

2.3. Sentiment Analysis

This paragraph provides an in-depth exploration of recent advances in sentiment analysis. In [49], the authors introduced the SentiDiff algorithm, which combines textual information with sentiment diffusion models to conduct sentiment analysis on Twitter data. Addressing the challenge of cross-domain sentiment coding using stochastic word-embedding techniques, ref. [50] proposes the CrossWord method. A unified framework, bridging machine learning and lexicon-based approaches, is presented in [51], where the authors introduced a genetic algorithm-based feature reduction technique, effectively addressing scalability issues. Moreover, ref. [52] introduced the SentiVec method, a kernel optimization approach for sentiment word embedding that integrates both supervised and unsupervised learning. By exploring various classifiers and feature sets for sentiment quantification, ref. [53] reveals the impact of different feature sets on classifier performance. For sentiment classification of online movie reviews, Ref. [54] uses the Bag of Words (BoW) technique and the Naive Bayes algorithm. In [55], a method combining feature extraction, the Word2Vec approach, and convolutional neural networks is proposed. Attention Emotion Enhanced (AEC)-LSTM, presented in [56], improves the LSTM network by incorporating an attention and emotional intelligence mechanism. Additionally, Ref. [57] introduces the Broad Multitask Transformer Network (BMT-Net), which enables learning global representations across tasks using multitask transformers. Finally, in [58,59], the BERT model is used to classify public sentiments regarding Covid-19.

2.4. Language Translation Models

One of the first language translation models based on machine learning employed an encoder–decoder recurrent neural network [60,61]. In [60], an extended version of the decoder that was able to leverage the context by exploiting an attention mechanism was proposed. Meanwhile, ref. [62] introduced a straightforward vocabulary substitution technique for adapting translation models to new languages, avoiding any architectural modifications. Other approaches employed the training of lightweight adapters [63],

language-specific encoder–decoders [64], and language-specific embeddings [65] to facilitate the learning of new languages. Recent approaches to language translation are based on continual learning, in particular, refs. [66,67] employed a method derived from Elastic Weight Consolidation [68] to alleviate catastrophic forgetting. Finally, the introduction of the Transformer model [8] radically changed the scenario in various natural language processing tasks, including translation. Indeed, model-based Transformers demonstrated the ability to achieve performance comparable to translations generated by humans [12,69].

3. Automated Moderator

To create an AM that can be used in place of an HM in a deliberative event, it is critical to understand the actions typically performed by the moderator. After an initial analysis, we concluded that the following list of features represents the minimum set of capabilities that should be integrated into the automated system to be effectively used in the deliberation process:

- **Welcome**: The AM is capable of detecting when a user joins a server, welcoming them, and providing necessary instructions. In the context of deliberation, this feature allows for sending each user the material needed for discussion, obtaining their informed consent, and providing guidance on the activity's progress;
- **Role Management**: The AM can assign and revoke roles (and, thus, the rights to perform certain actions) within the server. This enables the AM to manage conversation turns, giving users the opportunity to speak and/or write;
- **User Assistance**: Through the "help" command, users can request the AM to list its functionalities, providing details and explanations;
- **Scheduled Message Sending**: Scheduled messages, including both text and supplementary materials such as PDFs or images, can be scheduled to be sent by the AM at specific times. This feature guarantees programming the conversation agenda that is followed during the discussion by the AM. For example, in multi-topic discussions, one can plan the agenda by deciding when to move on to the next items. Furthermore, this also ensures that one can schedule follow-up messages that should be sent at a predetermined time after the end of a discussion session;
- **Promote interaction**: In case users remain inactive for a predefined period, the AM can automatically send a pre-configured message. The message is intended to introduce new discussion points and encourage users to actively participate in the conversation. This proactive approach ensures that the conversation remains dynamic and engaging;
- **Message Deletion**: The AM can delete user messages;
- **Kick User**: The AM can kick a user from the server, with the option for the user to rejoin later;
- **Ban/Unban User**: The AM has the authority to both expel a user and impose a ban, preventing them from joining the server unless the AM later lifts the ban;
- **Warn User**: The AM can issue and revoke warnings to users. Upon reaching a predefined warning threshold, the AM will automatically initiate a ban against the user;
- **Reporting Mechanism**: Using the AM, users can promptly report issues to facilitators who can intervene as needed. This reporting process ensures a quick way to solve problems that need the intervention of a human supervisor;
- **Event Logging**: The AM systematically records all events related to the discussion, including messages posted or deleted by users, warnings issued (along with corresponding reasons), and other relevant actions. This comprehensive event log provides a detailed record of discussion progress and user interactions that can be accessed at the end of the event for further analysis;
- **Translation Support**: To improve communication between users with different languages, the AM can autonomously translate user messages into various languages, facilitating the conversation;
- **Toxic Comment Detection**: The AM automatically analyzes all the messages posted by users to identify the use of toxic language. A message is considered toxic if the

content is rude, disrespectful, or alienating to someone from the conversation. After recognizing a message as inappropriate, the AM promptly removes it and sends a warning to the user who posted it. This proactive approach ensures that a respectful and inclusive conversation environment is maintained;

- **Spam detection**: The AM systematically analyzes all messages sent by users to identify spam. If a message is marked as spam, the AM immediately removes it and issues a warning to the user responsible for the message. This automatic spam detection mechanism helps maintain an orderly and productive communication environment;
- **Sentiment Analysis**: The AM analyzes each message and classifies it based on sentiment (positive or negative). This ability helps researchers gain a deeper understanding of the dynamics of discussion and the deliberative process.

We have implemented all the abovementioned features using the Python library discord.py. For features that required the AM to replicate intelligent human behavior, we leveraged AI tools via the PyTorch and TensorFlow libraries. Specifically, the translation support was implemented in PyTorch, while the toxic comment detection, spam detection, and sentiment analysis models were developed using TensorFlow. The training process of these models is detailed in the next section.

While other state-of-the-art AMs, as discussed in Section 2, focus mainly on content moderation, the proposed AM takes a more comprehensive approach by also effectively managing conversations, fostering interactions, and providing assistance to users when necessary. Furthermore, to the best of the author's knowledge, this AM is the first specifically designed for deliberative events and is also the first to integrate a translator to address the multilingual challenge. This enables seamless discussions between individuals from different linguistic backgrounds within a safe and inclusive environment. All features included in the AM were evaluated in a test event designed to validate the functionalities and the interaction between users and the AM. Specifically, we set an agenda and asked for the participation of five test users who tried to interact with each other on the server using different languages. Additionally, we intentionally encouraged users to use inappropriate language and send spam messages to evaluate the AM's effectiveness in detecting and removing such content and potentially banning users, if necessary. The test, even though it was not a real event, demonstrates the potential of the proposed AM in facilitating the deliberative process.

4. AI Tools

This section introduces AI tools integrated in the AM. Specifically, Section 4.1 outlines the main features of the translation system, while Sections 4.2 and 4.3 provide comprehensive insights into the dataset and the training process used for the toxic comment detector and the spam detector. Finally, Section 4.4 describes the sentiment analysis model and its training. It is important to note that, in deliberative events, the AM must operate in real time without interruption, preserving the fluidity of conversations. As a result, a delicate balance between performance and computation time had to be meticulously achieved in this study. This balance, necessary to make all AI tools work in parallel, guided the tool development process described in the following sections.

4.1. No Language Left Behind Translator

To facilitate and promote global connectivity, Meta AI researchers have launched "No Language Left Behind" (NLLB) [12], a global initiative focused on improving machine translation capabilities for a wide range of languages around the world. The NLLB model, based on Transformers, shows remarkable versatility by effectively translating content into 200 different languages. The main objective of the project is to ensure high-quality translation for languages that have not been adequately supported or completely neglected by current translation tools. The performance metrics of NLLB are impressive, with BLEU scores outperforming the previous state-of-the-art by an average of 44% in all 10,000 directions of the FLORES-101 benchmark [70]. This improvement is even more

pronounced, exceeding 70%, for some African and Indian languages, compared to recent translation systems. Meta has decided to open-source the NLLB model, encouraging researchers to expand its application to include even more languages, contributing to the development of more inclusive technologies. In this work, we used the distilled version of the NLLB model with 600 million parameters, accessible via the HuggingFace Transformers library (https://huggingface.co/docs/transformers/index, accessed on 15 June 2023). The translator is integrated with a language detection tool provided by the Spark NLP library (https://sparknlp.org/, accessed on 12 December 2023). When a user posts a message, the AM dynamically analyzes the text to identify the language and, leveraging the NLLB model, translates it into five different languages (Italian, English, French, German, and Polish). Finally, the translations are made available to all users by the moderator. This approach improves communication across diverse linguistic landscapes, fostering a more inclusive and interconnected digital environment.

4.2. Toxic Comment Detector

4.2.1. Jigsaw Multilingual Toxic Comment Dataset

The Conversation AI team provided the Jigsaw Multilingual Toxic Comment dataset, a research initiative established by Jigsaw and Google to develop technology to help create a safer and more collaborative Internet. The dataset was originally released for the 2020 Toxic Comment Detection Challenge (https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification/overview, accessed on 12 December 2023), and it is now publicly available on Kaggle. The dataset is divided into training, validation, and test sets, each including textual content and its classification as toxic or non-toxic. The training set includes only English comments from Civil Comments or Wikipedia talk page edits. In contrast, the validation and test datasets include comments in multiple languages (including Spanish, Italian, Turkish, Russian, Portuguese, and French). Table 1 reports the number of comments in each dataset split.

Table 1. Number of toxic and non-toxic comments in the three subsets splits.

Split	Num. Toxic Comments	Num. Non-Toxic Comments	Total
Training Set	94,084	202,156	296,240
Validation Set	1230	6770	8000
Test Set	14,410	49,402	63,812

4.2.2. Experimental Setup

Given the inclusion of multiple languages in both the original validation and test sets, we opted for a dual-model evaluation approach. The model was trained exclusively with English comments, using 85% of the original training set for training, while 5 and 10% of the reamainder served as the validation and test sets, respectively. Conversely, to evaluate the trained model in a multilingual context, we leveraged both the original validation and test sets, which included multiple languages. In this second scenario, before applying the classifier, the comments were translated into English using the translator described in Section 4.1. We used a classifier based on a distilled version of BERT [71], a neural network model based on Transformers. Transformers represent a significant advancement in deep learning, leveraging a self-attention mechanism that assigns varying weights to different segments of input data based on their relative importance. Thanks to this capability, these models prove to be particularly suitable for handling complex sequential data and find broad applications in Natural Language Processing. The pre-trained BERT model was obtained using the KerasNLP library. The final classification is obtained using a fully connected layer embedded on top of the BERT encoder. The BERT model requires tokenized input sentences for its processing; to this aim, we employed a pre-trained tokenizer based on the WordPiece tokenizer, which allows one to effectively

split a sentence into multiple tokens. The resulting textual representation is input to the model as a list of the IDs corresponding to each token.

The dataset outlined in Section 4.2.1 shows a significant class imbalance between the two categories (refer to Table 1). To address this issue, we implemented a translation-focused data augmentation strategy, specifically augmenting only toxic comments within the training set. To achieve this goal, we used the translator described in Section 4.1 to translate the toxic comments present in the training set into multiple languages, including Italian, English, French, German, and Polish. We subsequently translated these augmented comments back into English. This process produced sentences that retained their semantic meaning while employing different vocabulary. This approach serves a dual purpose: it mitigates the imbalance problem and integrates translated messages into the training process, preparing the classifier for evaluation in a multilingual setup.

4.2.3. Results

This section reports the results obtained by the model trained following the experimental setup described in Section 4.2.2 for toxic detection. In particular, Table 2 reports the results obtained on the validation and test sets containing only English comments. To mitigate the effect of class imbalance, the weighted average of the metrics reported in the table was calculated.

Table 2. Results on the validation and test set with English comments.

Split	Accuracy	Precision	Recall	F1-Score	AUROC
Validation Set	89.66%	93.25%	89.66%	90.85%	86.75%
Test Set	89.84%	93.29%	89.84%	90.99%	86.82%

The results on the multi-lingual validation and test sets are instead reported in Table 3.

Table 3. Results on the validation and test set with multi-lingual comments.

Split	Accuracy	Precision	Recall	F1-Score	AUROC
Validation Set	74.05%	86.63%	74.05%	77.47%	77.78%
Test Set	72.23%	82.51%	72.23%	74.45%	76.50%

As can be observed from the tables, it is clear that, in the English setting, the model presents a very good ability to correctly identify toxic comments. However, unfortunately, there is a significant drop in model performance when evaluated in the multilingual configuration. The results obtained show a notable deviation from the scores reported in the Kaggle competition leaderboard (https://www.kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification/leaderboard, accessed on 27 December 2023) (about 95% AUROC). However, it is important to highlight that a direct comparison of the proposed approach with the current leaderboard leaders is quite difficult, mainly for two reasons. First, since the competition is closed, the submission server is no longer available and it is impossible to calculate the results using the same data split used in the leaderboard. The Public score and the Private scores were computed, respectively, with 30% and 70% of the test data, and it is not possible to know which samples of the test set were included in each split. Second, our system is designed to run in real-time on a single NVIDIA 2080Ti GPU, which is the hardware available in the present study. For these reasons, it is unfair to compare our model with those not designed to operate without these limitations. In real-world test scenarios, the trained model demonstrated effectiveness, even with multilingual comments. However, to further improve its performance, incorporating additional training data labeled with multilingual comments would prove beneficial to the network training process.

4.3. Spam Detector

4.3.1. UtkMl's Twitter Spam Detection Dataset

The dataset used in this study comes from UtkMl's Twitter spam detection contest hosted on Kaggle (https://www.kaggle.com/c/utkmls-twitter-spam-detection-competition/overview, accessed on 12 December 2023). This dataset collects various types of spam messages, including automatically generated content, meaningless posts, and clickbait. All tweets in the dataset are written in English and, for each of them, information is provided about the text and the network of contacts of those who tweeted, along with other relevant details. Additionally, a label is included to indicate whether the tweet is classified as spam or not. An official training–test split is provided, and, within the training set, we created a validation subset (comprising 10% of the original set) to be used in the training process. It is important to note that the test set was created for the competition and, therefore, labels for this set are not publicly available. To evaluate models on this specific subset of data, the results must be submitted to the evaluation server. Table 4 reports the number of comments in each dataset split.

Table 4. Number of spam and non-spam comments in the three subset splits.

Split	Num. Spam Comments	Num. Non-Spam Comments	Total
Training Set	7443	5965	13,408
Validation Set	745	746	1491
Test Set	-	-	785

4.3.2. Experimental Setup

We used the same network architecture employed for toxic message classification (Section 4.2.2). Given the relatively balanced distribution between spam and non-spam classes in the UtkMl's Twitter Spam Detection dataset, we chose to not use data augmentation strategies to balance the dataset. The network was trained using the training set, while the validation set was used to stop early and avoid overfitting. Finally, the trained model was evaluated on the test set. When the model was integrated into the AM, we combined it with the translator (Section 4.1). This allowed messages written in languages other than English to be translated into English and analyzed by the spam detector.

4.3.3. Results

This section presents the results obtained by the model trained according to the experimental setup detailed in Section 4.3.2 for spam detection. The results obtained by the trained model on the validation set are reported in Table 5.

Table 5. Results on the validation set.

Split	Accuracy	Precision	Recall	F1-Score	AUROC
Validation Set	92.62%	92.35%	93.58%	92.96%	90.12%

To evaluate the model on the test set, we must use the evaluation server of the competition, which computes only the classification accuracy. In particular, on the test set, our model achieved a Public score of 90.63% and a Private score of 91.63%. Although the model was scaled for real-time performance on a single GPU, the results obtained are highly competitive when compared to the state-of-the-art reported in the Kaggle competition leaderboard (https://www.kaggle.com/c/utkmls-twitter-spam-detection-competition/leaderboard, accessed on 27 December 2023). In particular, our performance is particularly noteworthy when evaluating the Private score, lagging behind only the first two approaches. Instead, when considering the Public score, our approach guarantees a position in the top ten results. These results highlight the model's ability to accurately identify spam in

messages, making it suitable for effective integration into the AM. Unfortunately, the lack of multilingual comments in this dataset limits model evaluation to English only.

4.4. Sentiment Analysis

4.4.1. Large Movie Review Dataset

The Large Movie Review dataset [11] includes movie reviews, written in English, accompanied by binary sentiment labels, distinguishing between positive and negative sentiments. The reviews are collected from the Internet Movie Database, and the label is obtained using the score assigned by the user who wrote the comment. The dataset contains 50,000 reviews, split evenly into a training set (25,000) and a test set (25,000), with no overlap between the movies in the two sets. To mitigate potential bias resulting from related ratings, a maximum limit of 30 reviews per movie is enforced. Additionally, the reviews in the dataset are well balanced, with 25,000 positive and 25,000 negative comments. Given the absence of an official validation split, in this study, 10% of the training set was used for validation purposes.

4.4.2. Experimental Setup

We used the same network architecture employed for toxic message classification (Section 4.2.2), the only difference being that, for sentiment analysis, we did not need to apply any strategies to balance the dataset. The network was trained using the training set, while the validation set was used to stop the training early to avoid overfitting. Finally, the trained model was evaluated on the test set. When the model was integrated into the AM, we combined it with the translator (Section 4.1). This allowed messages written in languages other than English to be translated into English and to be analyzed by the trained model for sentiment analysis.

4.4.3. Results

This section presents the results obtained by the model trained according to the experimental setup detailed in Section 4.4.2 for sentiment analysis. In particular, Table 6 shows the results calculated on the validation and test set.

Table 6. Results on the validation and test sets of the model trained for sentiment analysis.

Split	Accuracy	Precision	Recall	F1-Score	AUROC
Validation Set	88.56%	89.92%	88.12%	89.01%	86.78%
Test Set	86.48%	88.23%	86.94%	87.31%	85.53%

For a comprehensive comparison of different sentiment analysis approaches on the large movie review dataset, please refer to the leaderboard (https://paperswithcode.com/sota/sentiment-analysis-on-imdb, accessed on 27 December 2023). There is, approximately, a 10% disparity in accuracy between our approach and the best results reported in the leaderboard. However, it is important to highlight that many state-of-the-art results are obtained using additional datasets and complex models, making a fair comparison with our approach difficult. The results are quite promising for the English language. Unfortunately, the lack of multilingual comments prevents the evaluation of the model in different languages.

5. Conclusions

This paper introduces an AM, which can be effectively used in place of an HM in deliberative events. Implemented in Python using the discord.py library, this moderator has been developed for the Discord online communication platform. The AM is equipped with AI tools, which allow it to perform a wide range of functions traditionally performed by HMs. In particular, three specialized deep learning models, trained for spam detection, toxic comment classification, and sentiment analysis, were included in the AM. These

models allow the AM to analyze user messages to identify and remove inappropriate content. The system sends warnings to users and, in the case of repeated violations of the rules, imposes bans to maintain the integrity of the discussion. In addition to content moderation, a built-in translator, leveraging the NLLB model, improves user engagement by facilitating communication in multiple languages. The effectiveness of the proposed AM has been validated through testing in simulated deliberative events, demonstrating its potential. One notable limitation is that AI tools are currently optimized for English, making it necessary to translate comments, originally written in other languages, before their analysis. Future research will address this constraint by training language-specific models to accommodate multilingual discussions. Furthermore, organizing a test in a real deliberative event to validate AM in a concrete scenario will also be the subject of further research.

Funding: This research was co-funded by the European Union—FSE REACT-EU, PON Research and Innovation 2014–2020.

Data Availability Statement: Information on the data used in the paper is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
BERT	Bidirection Encoder Representations from Transformers
NLLB	No Language Left Behind
AM	Automated Moderator
HM	Human Moderator
DL	Deep Learning
SVM	Support Vector Machines
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network

References

1. Boulianne, S. Building faith in democracy: Deliberative events, political trust and efficacy. *Political Stud.* **2019**, *67*, 4–30. [CrossRef]
2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
3. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 27: 28th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
4. Ching, T.; Himmelstein, D.S.; Beaulieu-Jones, B.K.; Kalinin, A.A.; Do, B.T.; Way, G.P.; Ferrero, E.; Agapow, P.M.; Zietz, M.; Hoffman, M.M.; et al. Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **2018**, *15*, 20170387. [CrossRef] [PubMed]
5. Bonechi, S. ISIC_WSM: Generating Weak Segmentation Maps for the ISIC archive. *Neurocomputing* **2023**, *523*, 69–80. [CrossRef]
6. Bonechi, S.; Bianchini, M.; Bongini, P.; Ciano, G.; Giacomini, G.; Rosai, R.; Tognetti, L.; Rossi, A.; Andreini, P. Fusion of visual and anamnestic data for the classification of skin lesions with deep learning. In Proceedings of the New Trends in Image Analysis and Processing–ICIAP 2019: ICIAP International Workshops, BioFor, PatReCH, e-BADLE, DeepRetail, and Industrial Session, Trento, Italy, 9–10 September 2019; Revised Selected Papers 20; Springer: Berlin/Heidelberg, Germany, 2019; pp. 211–219.
7. Bonechi, S.; Andreini, P.; Mecocci, A.; Giannelli, N.; Scarselli, F.; Neri, E.; Bianchini, M.; Dimitri, G.M. Segmentation of aorta 3D CT images based on 2D convolutional neural networks. *Electronics* **2021**, *10*, 2559. [CrossRef]
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
9. Otter, D.W.; Medina, J.R.; Kalita, J.K. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 604–624. [CrossRef] [PubMed]
10. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.

11. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
12. Costa-jussà, M.R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. No language left behind: Scaling human-centered machine translation. *arXiv* **2022**, arXiv:2207.04672.
13. Bruckman, A.; Curtis, P.; Figallo, C.; Laurel, B. Approaches to managing deviant behavior in virtual communities. In Proceedings of the Conference companion on Human Factors in Computing Systems, Boston, MA, USA, 24–28 April 1994; pp. 183–184.
14. Lessig, L. *Code: And other Laws of Cyberspace*; Basic Books, Inc.: New York, NY, USA. 2009.
15. Singh, S. Everything in moderation: An analysis of how Internet platforms are using artificial intelligence to moderate user-generated content. *New Am.* **2019**, *22*, 1–42.
16. Jhaver, S.; Bruckman, A.; Gilbert, E. Does transparency in moderation really matter? User behavior after content removal explanations on reddit. *Proc. ACM-Hum.-Comput. Interact.* **2019**, *3*, 1–27.
17. Kou, Y.; Gui, X.; Zhang, S.; Nardi, B. Managing disruptive behavior through non-hierarchical governance: Crowdsourcing in League of Legends and Weibo. *Proc. ACM-Hum.-Comput. Interact.* **2017**, *1*, 1–17. [CrossRef]
18. Jhaver, S.; Birman, I.; Gilbert, E.; Bruckman, A. Human-machine collaboration for content regulation: The case of reddit automoderator. *ACM Trans.-Comput.-Hum. Interact. (TOCHI)* **2019**, *26*, 1–35. [CrossRef]
19. Spertus, E. Smokey: Automatic recognition of hostile messages. In Proceedings of the AAAI/IAAI, Providence, RI, USA, 27–31 July 1997; pp. 1058–1065.
20. Razavi, A.H.; Inkpen, D.; Uritsky, S.; Matwin, S. Offensive language detection using multi-level classification. In Proceedings of the Advances in Artificial Intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010, Ottawa, ON, Canada, 31 May–2 June 2010; Proceedings 23; Springer: Berlin/Heidelberg, Germany, 2010; pp. 16–27.
21. Schmidt, A.; Wiegand, M. A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Valencia, Spain, 3–7 April 2017; pp. 1–10.
22. Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated hate speech detection and the problem of offensive language. In Proceedings of the International AAAI Conference on Web and Social Media, Montreal, QC, Canada, 15–18 May 2017; Volume 11, pp. 512–515.
23. Cox, D.R. The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B* **1958**, *20*, 215–232. [CrossRef]
24. Webb, G.I. Naïve Bayes. In *Encyclopedia of Machine Learning*; Sammut, C., Webb, G.I., Eds.; Springer: Boston, MA, USA, 2010; pp. 713–714. [CrossRef]
25. Wu, X.; Kumar, V.; Quinlan, J.R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Philip, S.Y.; et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [CrossRef]
26. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis And recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
27. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
28. Kennedy, G.; McCollough, A.; Dixon, E.; Bastidas, A.; Ryan, J.; Loo, C.; Sahay, S. Technology solutions to combat online harassment. In Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 73–77.
29. Wulczyn, E.; Thain, N.; Dixon, L. Ex machina: Personal attacks seen at scale. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 May 2017; pp. 1391–1399.
30. Djuric, N.; Zhou, J.; Morris, R.; Grbovic, M.; Radosavljevic, V.; Bhamidipati, N. Hate speech detection with comment embeddings. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 29–30.
31. Pavlopoulos, J.; Malakasiotis, P.; Androutsopoulos, I. Deep learning for user comment moderation. *arXiv* **2017**, arXiv:1705.09993.
32. Pavlopoulos, J.; Malakasiotis, P.; Androutsopoulos, I. Deeper attention to abusive user content moderation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 1125–1135.
33. Pavlopoulos, J.; Malakasiotis, P.; Bakagianni, J.; Androutsopoulos, I. Improved abusive comment moderation with user embeddings. *arXiv* **2017**, arXiv:1708.03699.
34. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
35. LeCun, Y.; Bengio, Y.; et al. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995.
36. Badjatiya, P.; Gupta, S.; Gupta, M.; Varma, V. Deep learning for hate speech detection in tweets. In Proceedings of the 26th International conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 759–760.
37. Del Vigna, F.; Cimino, A.; Dell'Orletta, F.; Petrocchi, M.; Tesconi, M. Hate me, hate me not: Hate speech detection on facebook. In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy, 17–20 January 2017; pp. 86–95.
38. Gambäck, B.; Sikdar, U.K. Using convolutional neural networks to classify hate-speech. In Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 85–90.
39. Park, J.H.; Fung, P. One-step and two-step classification for abusive language detection on twitter. *arXiv* **2017**, arXiv:1706.01206.
40. Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; Villata, S. A multilingual evaluation for online hate speech detection. *ACM Trans. Internet Technol. (TOIT)* **2020**, *20*, 1–22. [CrossRef]

41. Mathew, B.; Saha, P.; Yimam, S.M.; Biemann, C.; Goyal, P.; Mukherjee, A. Hatexplain: A benchmark dataset for explainable hate speech detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 14867–14875.
42. Roy, S.G.; Narayan, U.; Raha, T.; Abid, Z.; Varma, V. Leveraging multilingual transformers for hate speech detection. *arXiv* **2021**, arXiv:2101.03207.
43. Harisinghaney, A.; Dixit, A.; Gupta, S.; Arora, A. Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm. In Proceedings of the 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), Faridabad, India, 6–8 February 2014; pp. 153–155.
44. Laorden, C.; Santos, I.; Sanz, B.; Alvarez, G.; Bringas, P.G. Word sense disambiguation for spam filtering. *Electron. Commer. Res. Appl.* **2012**, *11*, 290–298. [CrossRef]
45. Jánez-Martino, F.; Fidalgo, E.; González-Martínez, S.; Velasco-Mata, J. Classification of spam emails through hierarchical clustering and supervised learning. *arXiv* **2020**, arXiv:2005.08773.
46. Liu, X.; Lu, H.; Nayak, A. A spam transformer model for SMS spam detection. *IEEE Access* **2021**, *9*, 80253–80263. [CrossRef]
47. Tida, V.S.; Hsu, S. Universal spam detection using transfer learning of BERT model. *arXiv* **2022**, arXiv:2202.03480.
48. Sahmoud, T.; Mikki, D.M. Spam detection using BERT. *arXiv* **2022**, arXiv:2206.02443.
49. Wang, L.; Niu, J.; Yu, S. SentiDiff: Combining textual information and sentiment diffusion patterns for Twitter sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 2026–2039. [CrossRef]
50. Hao, Y.; Mu, T.; Hong, R.; Wang, M.; Liu, X.; Goulermas, J.Y. Cross-domain sentiment encoding through stochastic word embedding. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 1909–1922. [CrossRef]
51. Iqbal, F.; Hashmi, J.M.; Fung, B.C.; Batool, R.; Khattak, A.M.; Aleem, S.; Hung, P.C. A hybrid framework for sentiment analysis using genetic algorithm based feature reduction. *IEEE Access* **2019**, *7*, 14637–14652. [CrossRef]
52. Zhu, L.; Li, W.; Shi, Y.; Guo, K. SentiVec: Learning sentiment-context vector via kernel optimization function for sentiment analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 2561–2572. [CrossRef]
53. Ayyub, K.; Iqbal, S.; Munir, E.U.; Nisar, M.W.; Abbasi, M. Exploring diverse features for sentiment quantification using machine learning algorithms. *IEEE Access* **2020**, *8*, 142819–142831. [CrossRef]
54. Khan, A.; Gul, M.A.; Zareei, M.; Biswal, R.; Zeb, A.; Naeem, M.; Saeed, Y.; Salim, N. Movie review summarization using supervised learning and graph-based ranking algorithm. *Comput. Intell. Neurosci.* **2020**, *2020*, 7526580. [CrossRef]
55. Kumar, R.; Pannu, H.S.; Malhi, A.K. Aspect-based sentiment analysis using deep networks and stochastic optimization. *Neural Comput. Appl.* **2020**, *32*, 3221–3235. [CrossRef]
56. Huang, F.; Li, X.; Yuan, C.; Zhang, S.; Zhang, J.; Qiao, S. Attention-emotion-enhanced convolutional LSTM for sentiment analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 4332–4345. [CrossRef]
57. Zhang, T.; Gong, X.; Chen, C.P. BMT-Net: Broad multitask transformer network for sentiment analysis. *IEEE Trans. Cybern.* **2021**, *52*, 6232–6243. [CrossRef]
58. Singh, M.; Jakhar, A.K.; Pandey, S. Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Soc. Netw. Anal. Min.* **2021**, *11*, 33. [CrossRef]
59. Aygün, I.; Kaya, B.; Kaya, M. Aspect based twitter sentiment analysis on vaccination and vaccine types in COVID-19 pandemic with deep learning. *IEEE J. Biomed. Health Inform.* **2021**, *26*, 2360–2369. [CrossRef] [PubMed]
60. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
61. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014.
62. Garcia, X.; Constant, N.; Parikh, A.P.; Firat, O. Towards continual learning for multilingual machine translation via vocabulary substitution. *arXiv* **2021**, arXiv:2103.06799.
63. Bapna, A.; Arivazhagan, N.; Firat, O. Simple, scalable adaptation for neural machine translation. *arXiv* **2019**, arXiv:1909.08478.
64. Escolano, C.; Costa-Jussà, M.R.; Fonollosa, J.A. From bilingual to multilingual neural-based machine translation by incremental training. *J. Assoc. Inf. Sci. Technol.* **2021**, *72*, 190–203. [CrossRef]
65. Berard, A. Continual learning in multilingual NMT via language-specific embeddings. *arXiv* **2021**, arXiv:2110.10478.
66. Thompson, B.; Gwinnup, J.; Khayrallah, H.; Duh, K.; Koehn, P. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. 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), Minneapolis, MN, USA, 3–5 June 2019; pp. 2062–2068.
67. Gu, S.; Feng, Y. Investigating catastrophic forgetting during continual training for neural machine translation. *arXiv* **2020**, arXiv:2011.00678.
68. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
69. Bapna, A.; Caswell, I.; Kreutzer, J.; Firat, O.; van Esch, D.; Siddhant, A.; Niu, M.; Baljekar, P.; Garcia, X.; Macherey, W.; et al. Building machine translation systems for the next thousand languages. *arXiv* **2022**, arXiv:2205.03983.

70. Goyal, N.; Gao, C.; Chaudhary, V.; Chen, P.J.; Wenzek, G.; Ju, D.; Krishnan, S.; Ranzato, M.; Guzmán, F.; Fan, A. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 522–538. [CrossRef]
71. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.