Pioneering Bot Detection on Polish Reddit at the Comment Level

Karmela Matyjaszek
University of Gdańsk, Poland
matyjaszek.karmela@gmail.com

Abstract

Research on bot detection in social media exhibits imbalance in several areas — across platforms, languages, and detection levels. Addressing these gaps, this study focuses on comment-level bot detection within Polish Reddit communities. We describe in detail the construction of a comprehensive dataset (∼40,000 comments, 58% bot-comment prevalence), which provides labels for the subsequent model training. Polish Reddit is inherently multilingual, we therefore take advantage of the linguistic signals, treating language composition of a comment as a feature on its own. We develop novel platform-specific, language-specific, and culturally informed features, and train comment-level classifiers from multiple model families on the manually annotated dataset. The resulting models achieve strong performance and temporal generalization to 2025 data. We analyze the importance and direction of these novel features across models and report that our 'cross-level' interaction features, 'Bottiquette' compliance signals, formatting markers, language indicators, repetition and randomness measures — especially the entropy of non-alphabetic characters — rank among the most decisive features. Finally, we complement our quantitative findings with a qualitative characterization of the Polish Reddit bot ecosystem. Overall, this study provides an important baseline for an underexplored setting and contributes to an open discussion on how to approach detection where data is linguistically mixed.

1 Introduction

The majority of empirical studies on bot detection are tailored to Twitter (Ng and Carley, 2022; Hurtado et al., 2019; Beskow and Carley, 2018a) and utilize its user-to-user network schema to extract features for model training (Beskow and Carley, 2018a). Meanwhile, Reddit presents an entirely different type of social network that is primarily topic-based (Hurtado et al., 2019). This renders most Twitter-tuned approaches inapplicable to Reddit and other social media websites that do not rely on user-to-user subscriptions. At the same time, hardly any studies have been dedicated solely to Reddit bot detection.

Although platform-agnostic approaches do exist, their broader scope often comes at the expense of optimal performance (Ng and Carley, 2022; Yang et al., 2019b). In the case of Reddit, the lack of specificity may be particularly undesirable: our study suggests that a substantial portion of its bot activity tends to follow simple patterns that roughly align with its 'Bottiquete' recommendations[1]. While these patterns can be used to extract numerous useful features, they are perhaps too platform-specific and at present have not yet been incorporated into multi-platform frameworks. In contrast, our work does not attempt to cover a range of social media websites, but instead targets Reddit exclusively, and therefore can freely make use of such features.

While the number of publications discussing detection in non-English contexts is increasing, such settings are still underexplored and remain particularly challenging. Studies that devise dedicated solutions for languages other than English often report their lower performance in settings that enable comparisons, like the PAN at CLEF 2019 task (Bacciu et al., 2019; Pizarro, 2019). Rauchfleisch and Kaiser (2020) show that applying even a well-established system (Botometer) to non-English data may result in degraded accuracy. Non-English bot detection is often overlooked (with the field being largely English-centric), or artificially separated from 'general' detection. This separation occurs both at the level of data selection and training (e.g., the PAN at CLEF task providing separate English

[1] https://www.reddit.com/r/Bottiquette/wiki/bottiquette/, referenced also by Hurtado et al. (2019)

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 881–894

and Spanish data) and at the level of real-world detection (e.g., the earlier version of Botometer requiring the end user to specify the language of the targeted account so that the separate language-agnostic classifier could be picked for non-English data).

In our work, we view the Polish context as an opportunity, a perspective reflected in our feature selection, feature engineering, and other methodological choices. We do not filter our data by language; instead, we restrict our sources to subreddits where Polish culture is expected to dominate, whether through language choice or specific cultural markers (such as the 'XD' usage discussed later), with the full awareness that the resulting data will be linguistically mixed. In this sense, our study should be conceptualized less as "language-specific" bot detection and more as detection within a setting dominated by a specific cultural group. This population is inherently multilingual: Reddit is a global platform, and many Polish-focused communities either openly welcome English speakers (e.g., r/poland) or utilize English as a lingua franca due to the subreddit's purpose (e.g., r/learnpolish). In fact, 42% of words within our dataset have been identified as English.

Crucially, we do not treat the language of a comment as a passive background variable or a filter, but rather as an active component of the user's behavioral profile. We integrate language choice directly into our feature engineering, recognizing that a comment's language or whether a user switches between languages is a signal in itself. This distinguishes our approach from prior work; for instance, while Stukal et al. (2017) targeted Russian bots on Twitter using keywords that appeared in both Latin and Cyrillic scripts, they did not explicitly operationalize script usage as a feature (e.g., a "Cyrillic usage" metric). In contrast, we explicitly model this behavior by including features such as English word count, Polish word count, Polish density (i.e., language ratio), and diacritics usage, allowing the model to interpret the specific linguistic composition of a message as part of the detection logic.

Furthermore, for practitioners whose primary goal is to clean text corpora for downstream NLP tasks, the prevailing bot detection frameworks can be ill-suited. Driven primarily by concerns about online manipulation, misinformation, and coordinated inauthentic behavior across large social media platforms (e.g. Stukal et al., 2017; Yang et al., 2019a; Cresci, 2020) they often require extensive profile crawling or network analysis (Beskow and Carley, 2018a), introducing dependencies that fall outside the scope of typical text-oriented workflows. There is limited guidance on how to perform efficient bot filtering when bot detection is not the object of study but one of several preprocessing steps within a larger NLP pipeline, where full account screening may not be feasible.

To address the scarcity of comment-level research (Yang et al., 2019b; Kudugunta and Ferrara, 2018) and the practical limitations of full account screening (Kudugunta and Ferrara, 2018; Beskow and Carley, 2018a), we develop *comment-level* detection models using single-comment features, which include language-specific, culture-specific, and platform-specific features, as well as various stylometric features, often marking the level of repetition within a comment. Given that bot evolution over time often compromises detection reliability (Varol et al., 2017; Ng and Carley, 2022; Yang et al., 2023), we also validate our models against temporal shifts. We then interpret the decision-making process of our classifiers using SHAP values to provide insight into the contribution of our novel features (Fig. 1 shows an example SHAP summary with top 30 features of the XGBoost model). Our dataset consists of 40,791 manually annotated comments from 1,418 users in Polish subreddits.

The study concludes with a qualitative profile of Polish Reddit bots, offering the first descriptive analysis of this ecosystem to guide future detection efforts.

2 Related Work

Hurtado et al. (2019) represents the first study focused exclusively on Reddit bot detection, specifically targeting coordinated influence operations. They adopted a network-based methodology to discover users acting in concert. By constructing a graph where connections were defined by the co-commenting behavior of users, they identified clusters of hyper-connected accounts within a previously chosen subreddit. They then corroborated the structural findings with temporal analysis.

Hurtado et al.'s study did not utilize a labeled ground-truth dataset; instead, they noted that the central nodes of the graph exhibited inhumanly low downtime between comments and that many of the detected high-connectivity accounts contained a *bot* substring in their usernames. In our own work, we confirm the ubiquity of automated accounts

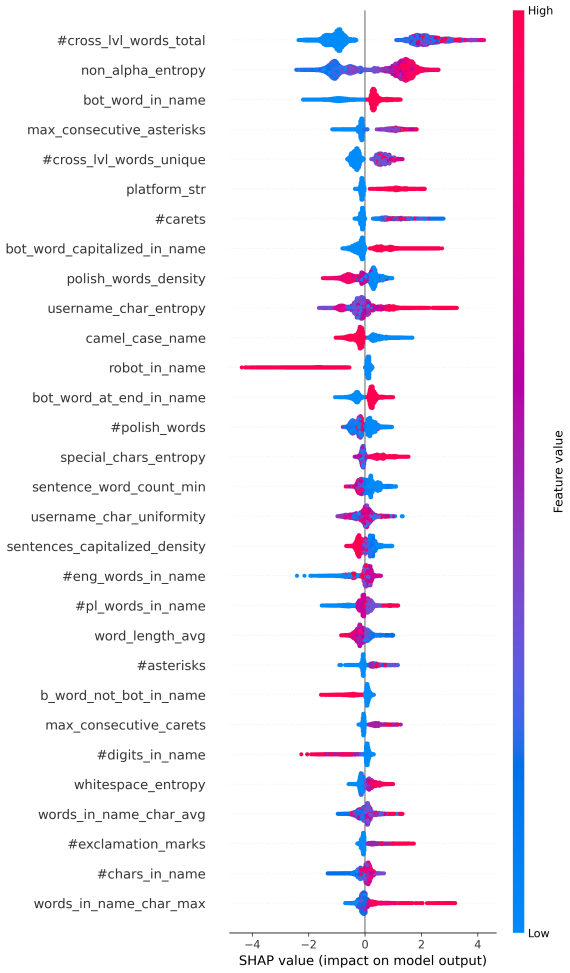

Figure 1: SHAP Summary of the 'CV' XGBoost.

that explicitly self-identify via their usernames (i.e., adhering to Bottiquette).

In multi-platform bot detection, few systems explicitly use Reddit training data. Adel Alipour et al. (2023), comparing their platform-independent approach against Twitter-centric models like Botometer, discuss the difficulty of transferring features across platforms. Their solution is to use strictly temporal features of user activity. An important caveat, however, is that their Reddit training data is derived entirely from the Russian troll list released by Reddit in 2017. Since troll accounts are often human-operated — and even automated ones exhibit specific behavioral patterns — this dataset offers a restricted scope of characteristics that may fail to support the generalizability the study seeks.

Ng and Carley (2022, 2024) adopted a different approach for their BotBuster series, relying instead on the crowdsourced BotRank list to establish the ground truth for Reddit bots. They specifically address the problem of information-constrained bot detection, proposing a method designed to function under incomplete account data conditions. They utilize many of the features we use in our own study, such as word and character counts or entropy, although reporting accuracy of only 35.68% on the Reddit dataset with an ensemble model.

Stieglitz et al. (2017) highlighted that the vast majority of bot detection techniques are designed for the English-speaking Twitter, leaving other linguistic spheres largely unmonitored. Among the language-agnostic or multilingual detection systems, the focus is typically to minimize or eliminate reliance on semantic content, favoring metadata and structural patterns that remain consistent across languages. Knauth (2019), for example, achieved high performance using language-independent features such as posting frequency, username length, and follower ratios. Within the shared PAN at CLEF 2019 task, which challenged participants to detect bots in both English and Spanish, top-performing teams like Pizarro (2019) utilized character n-grams and structural stylometry to bridge the language gap. Varol et al. (2017), observing that non-English tweets degraded their initial classifier performance, developed a dual-feature system: one set tailored for English data and a second comprising language-independent features for non-English content. Rauchfleisch and Kaiser (2020) show that that even well established, 'universal' bot detection systems can struggle with non-English data.

Our work demonstrates that successful detection in a multilingual setting without language separation or silencing language signals is achievable across multiple model families. We join Hurtado et al. in their focus on Reddit but utilize a ground-truth dataset and supervised learning techniques. While most prior work prioritizes author-level detection, in our view, it is valuable to gather a deeper understanding of how predictive features and models behave at the finer, comment-level granularity, with the added practical utility of supporting data cleaning pipelines.

3 Methodology

3.1 Terminology

The distinction between the terms 'bot' and 'social bot' remains ambiguous in the literature. The field lacks a unified definition, with individual studies often constructing *ad hoc* classifications tailored to specific datasets or disciplinary lenses (e.g., technical automation vs. sociological impact) (Cresci,

2020; Stieglitz et al., 2017; Gorwa and Guilbeault, 2018). Even comprehensive reviews diverge on this issue: Grimme et al. (2017) treats 'social bot' as a broad superordinate category encompassing various automated agents, whereas Stieglitz et al. (2017) argue for a narrower definition, asserting that "not every bot on social media is a social bot." They reserve the latter term specifically for programs designed to mimic human behavior and engage in social interaction.

In this work, we adopt the taxonomy proposed by Stieglitz et al. (2017), treating the term 'bot' as the overarching category. Consequently, we conceptualize our task as the detection of Reddit bots in general, a superset that may include social bots but is not limited to them. Given our interest in the practical utility of our systems, we forego complex network reconstruction or interaction monitoring typically required to profile complex social bots, including human-like yet automated trolls.

3.2 Data

We began by collecting comments posted until the end of November 2025 across subreddits that primarily or partially utilize the Polish language, yielding an initial pool of $\sim$10 mln comments with metadata such as the timestamp, author's username, and vote score. From this pool, we constructed a study dataset by selecting comments that satisfied at least one of the following criteria:

(1) *The comment body contains variations of both 'm' and 'bot' strings.* This captures English declarations like "I'm a bot" as well as Polish equivalents such as "Jestem botem".

(2) *The comment body contains variations of the phrase 'by a bot'.* This targets standard disclosures compliant with Bottiquette.

(3) *The comment body contains combinations of 'beep', 'boop', 'bleep', or 'bloop'.* It is a common humorous convention used by automated accounts to signal their nature.

(4) *The author's username contains the substring 'bot'.*

These filters were selected with the assumption that they would effectively capture a significant volume of overt, non-malicious bots that adhere to established Reddit conventions rather than concealing their identity. Simultaneously, we anticipated that these loose matching criteria would also retrieve (1) a substantial number of human users that happened to use similar strings, thereby naturally forming the 'human' portion of our dataset, and (2) a smaller number of less transparent bots.

After removing 48 duplicate entries caused by platform-side processing errors and 4 comments that shared the string '[deleted]' as the username, the dataset includes 1,418 authors and 40,791 comments. Bot authors represent 29% of all distinct accounts (411 out of 1,418). Those of their comments that were labeled as bot-generated make up 58% of all comments (23,662 bot vs 17,129 human comments). Four accounts exhibited mixed activity, producing both human- and bot-labeled content.

3.3 Annotation

We designed a two-stage annotation protocol that generates class labels while capturing additional metadata to enhance the dataset's utility.

Internal Stage: In the first stage, the annotator reviews comments of a single author and has access to comments' bodies (texts), timestamps, and the author's username. Based on this evidence, they classify the account as 'bot' or 'human' and assign an initial confidence score (range 1-3, from least to most confident).

External Stage: In the second stage, the annotator validates their assessment by consulting external sources — they are directed to the author's live Reddit profile to gather additional context. If an account is inaccessible (suspended or deleted), the annotator is instead directed to a platform-wide search for the username to leverage residual evidence, such as discussions by other users. They then provide a final classification and confidence score.

The distinction between human and automated accounts is often fluid — users may deploy personal accounts for automation or intervene manually on bot accounts (Chu et al., 2010; Varol et al., 2017; Cresci, 2020). Thus, we established specific classification criteria: *classify an account as a bot if any comments in the dataset exhibit automated characteristics, regardless of occasional human activity.* Although external validation revealed substantial mixed activity in live profiles, only four authors showed mixed behavior within our dataset comments. These cases underwent granular per-comment labeling, eventually reassigning 16 human-like comments from a bot account to the human class for comment-level training. A note on a labeling decision regarding an instance of possible bot-human mixing within one comment, which the labelling procedure had not accounted for, can be found in Appendix A.

The internal stage mirrors the information horizon available to our models (text, username, timestamps), enabling direct human-vs.-model comparisons of "classification without external context". However, final ground truth labels were established via the external stage, which interprets dataset evidence in light of broader account activity, making it more suitable for training.

The external stage also captures supplementary metadata: (1) account status (*exists/banned/deleted by user*), (2) whether there is residual evidence for inaccessible accounts (such as discussions), and (3) whether there is evidence of mixed activity not limited to the dataset. The latter applied only to bot-labeled accounts, as no human-labeled accounts in our dataset exhibited external bot activity.

3.4 Wordlists

The language-specific features rely largely on our English and Polish wordlists. The former is primarily derived from the NLTK API[2] (473,465 unique words); the latter (5,456,057 unique forms, including the inflected forms) — from the Polimorf dictionary of Morfeusz 2[3], a well-established Polish inflectional analyzer and generator. We also used a few smaller, custom English and Polish wordlists.

To account for for the prevalent social media practice of omitting diacritics for convenience (e.g., typing *patrzec* instead of *patrzeć — to look*), we expand our lexicon with de-diacritized variants. This involves generating a fully normalized ASCII version for each word with diacritics, as well as computing all combinatorial permutations of partial de-diacritization, the latter representing all possible combinations where users might omit only a subset of diacritics (i.e., capturing inconsistent typing). Fig. 2 illustrates the size of these derived sets.

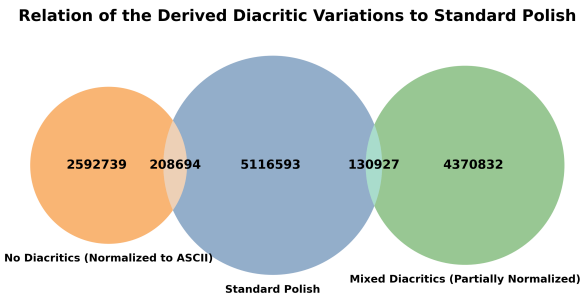

Figure 2: Set overlap of the standard Polish wordlist and its fully and partially de-diacriticized variants.

[2] https://www.nltk.org/
[3] https://morfeusz.sgjp.pl/

3.5 Feature Engineering

For feature extraction, we restricted our metadata usage to (1) timestamps, (2) usernames, and (3) *banned* flags, combining these with text-based features to create a multi-level but still lightweight feature set of 140 items.

We extracted binary indicators (e.g., presence of specific tokens in the username or comment body) and calculated various statistical measures. For text and username features, these included simple counts (e.g., URL count, character count), ratios (e.g., whitespace density), and more complex metrics such as Shannon entropy (character- and word-level). At the character level, these were, for instance, the entropy of whitespace, non-alphabetic, or special characters. We also derived three features from the timestamp metadata (cyclical hour-of-posting, day-of-the-week, and is-weekend) and included the *banned* flags, suspecting that bots may get suspended more often than human users.

3.5.1 Cross-Level Interaction Features

A critical component of our feature engineering involves cross-level features that capture the interplay between the username and the comment text. Unlike features confined to a single modality, these metrics identify when substrings from the username appear within the comment body. For example, if a bot named 'videobot' uses the word 'video' in a comment, this generates non-zero values for corresponding binary flags, frequency counts, and prevalence ratios. Our analysis indicates that these interaction features are among the most decisive predictors for correct classification.

To construct these features, we first tokenized the comment text to extract a set of priority terms. These were then merged with supplementary English and Polish word lists to form a comprehensive candidate vocabulary for username matching. To segment usernames — which typically lack explicit delimiters — against this vocabulary, we employed greedy dictionary matching, where the algorithm iteratively selects the longest matching substring from the wordlist.

Having segmented the usernames, we identified most frequent words within them, excluding 'bot'. Unsurprisingly, these primarily captured incidental 'bot' string matches such as 'Both', 'Bottle', or 'Bottom'. We therefore created a feature that accounts for the presence of words within the username that *start with 'b' and are not 'bot'*, plus a feature targeting the word 'robot'.

3.5.2 Polish-Specific Features

Polish-specific features primarily comprise Polish-language occurrence statistics derived from the Polish wordlists, including Polish words density within a comment, which is often a fraction between 0 and 1 due to code-mixing. 'Potential diacritics' count captures non-standard typing of possibly Polish words. While Morfeusz 2 could have enabled richer features such as part-of-speech tags, we prioritized computational efficiency and adopted a lightweight dictionary-based approach.

There is a culture-specific phenomenon in Polish whereby 'XD' is used both as an emoticon — arguably more frequently than in many other languages — and as a spoken expression pronounced out loud. It was selected as the Polish Youth Word of the Year 2017 in the PWN competition. Despite its popularity, there is relatively little linguistic work on Polish 'XD' usage. An exception from this is Kapuścińska (2020) who, debating whether emoticons can be treated as linguistic signs, discusses how 'XD' functions in the Polish language usage and shows that in Polish it has effectively attained the status of a word.

Given the perceived prominence of 'XD' in Polish and under the assumption that automation tends to favor more generic expressive markers, features capturing 'XD' usage (counts, densities) were included as potential human indicators. While analogous features were also constructed for other emoticons and emojis, these are not tied to Polish language and culture, and — being less localized — were expected to be more evenly distributed between human and automated accounts.

3.6 Word-Level Language Identification

Identifying whether individual words are Polish or English is a prerequisite for our language-specific features. Unfortunately, existing literature offers little guidance here: most studies on non-English bots do not specify their language identification method or rely on coarse metadata. Even standard tools like FastText operate at the comment level and struggle with the code-mixing (switching between languages), common in our dataset. Instead, we rely on a dictionary-based approach, matching tokens against our Polish and English wordlists.

The main challenge with this method is vocabulary overlap — 14,745 English words also appear in our Polish lexicon. This ambiguity increases when diacritics are removed: 17,072 English words overlap with the de-diacritized Polish words.

To resolve these ambiguities, we apply a simple context heuristic. For every comment, we first count the words that are uniquely Polish or uniquely English. We then classify any ambiguous words as belonging to the language that is already dominant in that specific comment.

3.7 Model Training

We performed a 5-fold grid-search cross-validation over several linear and tree-based classifiers: Logistic Regression, Naive Bayes, Support Vector Machine, Random Forest, Gradient Boosting, and XGBoost. We refer to them as 'CV models'. To ensure comparability, all models used the same stratified and shuffled 5-fold train-test split of data that was temporally restricted to December 2024 or older to enable the subsequent temporal robustness testing. The 0.2 train-test ratio we used yielded 25,842 observations in the training set and 6,461 in the test set (60.06% bots).

For each classifier, we selected the best *balanced* configuration, defined as the model achieving the highest mean performance across evaluation metrics within its grid-search results. These selected models were then retrained on data available up to December 2024 and evaluated month by month on 2025 holdout data (January–November) to assess temporal robustness. We refer to these as '2025 models'. This timestamp-based split yielded 32,303 comments used for training and 8,484 for evaluation, and was not stratified to preserve the natural shift in class distribution observed in the 2025 data, thereby providing a more authentic evaluation of real-world performance. As such, the test set is characterized by a substantial shift in bot activity (distribution change from $\sim$60% to $\sim$50% bot prevalence) — the so-called concept drift.

A table listing the exact counts and bot/human ratios for each split and fold can be found in Appendix B. Hyperparameters of all 2025 and CV models are listed in Appendix C.

3.8 Feature Analysis

We chose feature importance analysis with SHAP values (Lundberg and Lee, 2017) for two main reasons. First, compared to methods like Permutation Importance, it is more robust to the collinearity of features — a problem common in text classification, also present in our study. Second, it is model-agnostic. The latter is important because we analyze a range of models, which, we hope, will

provide some initial broad insight into how different model families handle bot detection within Polish Reddit communities. At the same time, defining "feature importance" in the multiple-model analysis setting is problematic (see Appendix D). Most importantly, the composition of feature importance is different in each model. We thus employ a variety of techniques to inform our understanding in this matter. We experiment with ranking features by the normalized mean SHAP value, by simple rank frequency, and by Reciprocal Rank Fusion (RRF) (Cormack et al., 2009) with the smoothing factor $k = 60$, as in the original study. We also visually inspect beeswarm plots for the top 30 most important features of each model, such as Fig. 1. Each technique highlights a different aspect of the cross-model feature importance (see Fig. 5).

The sign of a feature's mean SHAP value (+/-) is not a reliable indicator of the direction of its influence on model predictions. For example, the Polish word count feature has a positive mean SHAP value in our logistic regression models, however, interpreting this as evidence that the feature favors the positive (bot) class would be incorrect; in reality, higher values of this feature push predictions toward the human label. The positive mean arises from the prevalence of English-speaking bots in our dataset, which produces a large mass of observations with #polish_words = 0, thereby skewing the average. To interpret the directional impact of features across heterogeneous model architectures, we analyzed the slope of the SHAP dependence plots. We performed a linear regression on the feature-value/SHAP-value pairs for each feature-model combination. A positive slope indicates that increasing the feature value drives the model prediction towards the positive (bot) class.

4 Results

4.1 Model Performance

Table 1 summarizes the performance of CV and 2025 models. Tree models provide the best performance, with the XGboost model scoring the highest in *all* metrics in the CV testing, and in *most* metrics in 2025 testing. The score differences between the tree models are marginal. Interestingly, the Gradient Boosting model achieved its high scores with the subsampling parameter set to 1.0, which means no subsampling (as dictated by the grid search).

SVM has CV scores almost as high as the tree-based models, however, in 2025 testing, it scores lower, more similarly to the Logistic Regression models, which in turn still perform better than the Naive Bayes models. In general, all models exhibit strong performance and temporal generalization, with only a minor degradation in September–November (see Fig 3).

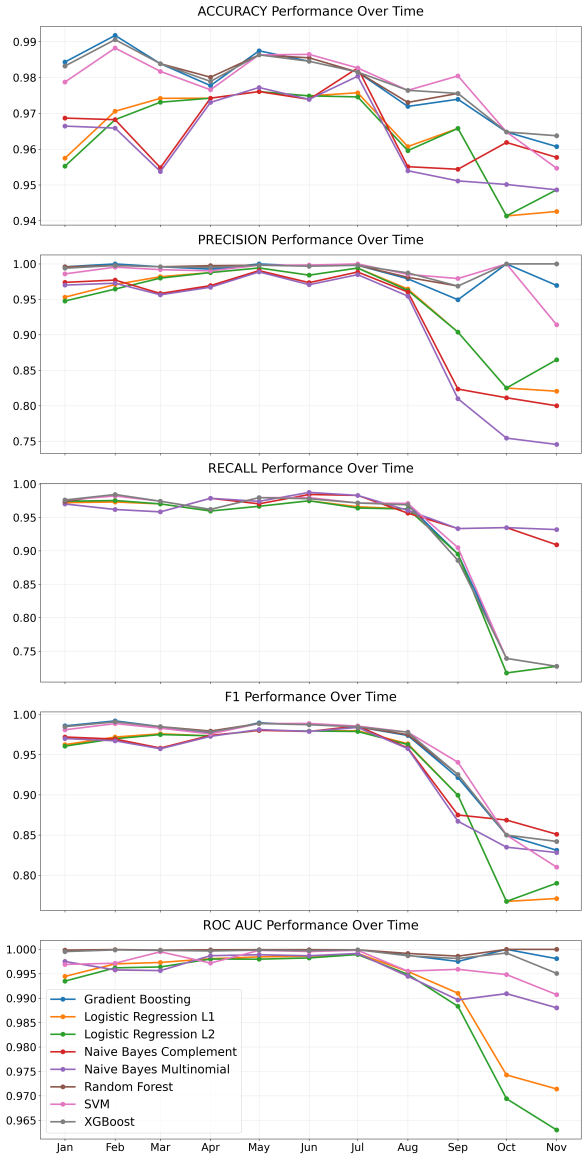

Figure 3: Monthly performance of ≤2024 (CV) models on 2025 holdout data.

Recall scores are lower than precision scores, regardless of model type or run type (i.e., CV/2025) (Tab. 1). However, while the Naive Bayes's performance is worse than that of other models (e.g., it is the only model whose CV recall score drops below 0.9), it is also the only model whose 2025 scores *improve* compared to its respective CV scores. Something in this model reacted positively to the distribution shift, while the rest of the models, as expected, experienced a small performance drop when tested

Model	F1		ROC AUC		Accuracy		Precision		Recall	
	CV	2025	CV	2025	CV	2025	CV	2025	CV	2025
XGBoost	**0.9980 (0.0007)**	**0.9809**	**0.9999 (0.0001)**	0.9996	**0.9976 (0.0009)**	**0.9811**	**0.9982 (0.0006)**	**0.9949**	**0.9978 (0.0011)**	0.9673
Random Forest	0.9975 (0.0005)	0.9808	0.9998 (0.0001)	**0.9998**	0.9970 (0.0006)	0.9810	0.9978 (0.0003)	0.9949	0.9972 (0.0008)	0.9671
Gradient Boost.	0.9978 (0.0003)	0.9805	0.9999 (0.0001)	0.9996	0.9973 (0.0004)	0.9807	0.9979 (0.0005)	0.9937	0.9977 (0.0008)	**0.9676**
SVM	0.9969 (0.0008)	0.9618	0.9992 (0.0003)	0.9889	0.9963 (0.0009)	0.9629	0.9976 (0.0010)	0.9942	0.9963 (0.0012)	0.9314
Log. Reg. L1	0.9890 (0.0007)	0.9665	0.9982 (0.0009)	0.9955	0.9868 (0.0009)	0.9669	0.9895 (0.0014)	0.9823	0.9885 (0.0021)	0.9511
Log. Reg. L2	0.9888 (0.0006)	0.9652	0.9982 (0.0009)	0.9965	0.9866 (0.0007)	0.9655	0.9893 (0.0015)	0.9771	0.9884 (0.0020)	0.9535
N. Bayes Comp.	0.9254 (0.0037)	0.9424	0.9810 (0.0010)	0.9728	0.9154 (0.0039)	0.9428	0.9833 (0.0008)	0.9520	0.8740 (0.0066)	0.9330
N. Bayes Mult.	0.9268 (0.0047)	0.9425	0.9811 (0.0010)	0.9727	0.9166 (0.0050)	0.9428	0.9801 (0.0017)	0.9514	0.8789 (0.0078)	0.9337

Table 1: CV and 2025 Model Performance Across Metrics. The best values are bolded, SD — in parentheses.

on the temporally distant 2025 data. The model also stands out in Sep–Nov scores (Fig. 3), handling the shift in the 2025 bots' characteristics more gracefully than otherwise better performing models. The ensemble models have better 2025 precision scores compared to recall, although the results are still lower than the CV precision.

4.2 Feature Importance and Direction

Cross-level interaction features (username-comment word overlap), Botiquette compliance signals, formatting markers (esp. carets count), entropy features (esp. non-alphabetic characters entropy), repetitivness measures (esp. n-gram repetition ratios) and language signals consistently ranked as important predictors across models and time periods. In contrast, emoticon- and emoji-based features, temporal features, and binary *banned* indicators were generally assigned low importance by the models. Features related to "XD" usage were also considered uninformative in most cases, despite being more common in human comments. Cross-model feature importance plots can be found in Appendix D.

CV models unanimously agree on the direction of 53 features, while the 2025 models — on 57 features. Combined, both fully agree on 45 features (30 bot, 15 human). Three out of four cross-level features are unanimously seen as bot signals: language-agnostic counts of cross-level words (total/unique) and the counts of English (but not Polish) cross-level words. Furthermore, the number of carets, asterisks, and exclamation marks, as well as whitespace entropy and punctuation entropy are clear bot markers. Conversely, the number and density of Polish words, the number and density of 'XD', as well as the presence of certain words in the username, like 'robot', are human indicators. For more details on the directional consensus — including the 45-item 'perfect-consensus' feature list — see Appendix E. The full feature list and descriptions can be found in Appendix F.

We observe that linear models favor n-gram uniqueness and diacritics count, as well as other counts that signal meticulous formatting or URL presence (the number of forward slashes). Ensemble models favor entropy, uniformity, and density based features. Computing Spearman correlation of the RFF rankings for Logistic Regression, Naive Bayes, SVM, and tree-based models (four groups) shows that Logistic Regression's rankings exhibit high similarity to SVM (0.75) and slightly lower to the Naive Bayes models (0.67), and that the rankings of the tree-based models are vastly different from the Naive Bayes models (0.31).

5 Discussion

Many features employed in this study have clear precedents in the existing literature. URL-based indicators and basic statistical counts have been used since the earliest social bot detection studies (Yardi et al., 2010). Temporal features are also common, particularly in platform- or language-agnostic systems (Chavoshi et al., 2016; Knauth, 2019; Adel Alipour et al., 2023), and username-based features have gained increasing attention in recent work (Beskow and Carley, 2018b; Yang et al., 2019b; Ng and Carley, 2022). However, the features that our models found the most important, such as cross-level interaction features, represent a relatively novel contribution to the bot detection feature space. Creating several entropy features targeting different types of characters (non-alphabetic, whitespace) also turned out to be beneficial.

While high Polish word counts and Polish ratio are strictly human, the counts of *potentially* diacriticized words were seen as bot signals, and the actual diacriticized words — as mixed. The role of language signals is multifaceted rather than simple and warrants further exploration in future studies.

6 Characterization of Polish Reddit Bots

Polish Reddit hosts predominantly benign or utility bots that tend to adhere to Botiquette and openly

admit they are automated. A substantial proportion announced their status or purpose either in comments or through usernames.

We initially hypothesized that inaccessible (suspended or deleted) profiles classified as bots during the internal evaluation stage would have generated extensive discussions that we would encounter in platform-wide searches. Instead, results revealed more nuanced patterns. Very concise notices appeared on r/BotWatch, a subreddit dedicated to community moderation through voting-based account banning. However, actual human discussion often surrounded bots that, although now suspended, had been genuinely appreciated by some users. These users created posts expressing that they missed or had enjoyed the functionality of the banned bots, revealing positive sentiment toward certain automated accounts we did not anticipate.

During the analysis of the accessible 'cyborg' profiles, we observed that bot developers frequently used these accounts for administrative purposes, including technical problem notices, usage instructions, shutdown announcements, and sharing tips for running Reddit bots or open-source code. A few bots maintained dedicated subreddits, likely created by their developers for focused interaction or testing. Some exhibited human-like activity unrelated to running bots, where operators occasionally posted casually as a regular profile.

Polish Reddit users are aware of bot presence on the platform and engage in playful mimicry of bot signaling conventions. Common examples include ironic "beep boop" declarations or replicating reply patterns of popular spellcheck bots when manually correcting other users' typos. Annotators need to be careful not to mistakenly classify such comments. Many non-bot matches in our dataset originate from human meta-commentary: accusing others of being bots during heated political discussions, self-defense claims, or participation in bot-related conventions like "Good bot" upvotes rewarding useful bots. Developers commonly append opt-out acknowledgments or feedback requests ("Good bot/bad bot"). Some bots were programmed to humorously subvert this by replying "good human" or "bad human" to users, while humans occasionally replied to disliked human comments with the "bad bot" phrase to express disapproval.

A distinctive subculture involved "bot wars" — developers creating accounts to combat other bots, frequently incorporating "anti-" prefixes (e.g., "anti-targetbot-bot"). These countermeasures occasionally spawned recursive opponents designed to neutralize the anti-bot measures themselves. Some exhibited apparent vengeful motivations, with developers accusing rival bots of plagiarism, inferiority, or being unauthorized copies.

Disruptive bots primarily manifested as spam accounts. However, one particularly illustrative incident preserved in a search results discussion involved a suspended bot that had persistently targeted a specific user across threads with unwanted replies, which resulted in the user's complaints.

More challenging to detect were bots employing scripted repetition of quotes from films or television series. When context was sparse (only a few comments), these seemed like human, albeit chaotic, responses. Internal detection depended heavily on the annotator's personal familiarity with the source material — when recognized, identification was straightforward; when unfamiliar, the repetitive nature only became apparent through external profile review. In other words, human annotation both benefits from outside world knowledge and remains vulnerable to knowledge gaps. Automated detection can perhaps compensate through full account analysis (revealing repetition patterns) or signals invisible to humans, such as network connections or behavioral metadata, but in supervised approaches it still depends on the quality of labels, which themselves may be flawed due to lapses of human judgment. Implementing solutions that detect cultural and political references could plausibly increase both human annotation and model bot detection reliability.

7 Conclusion

Our comment-level study is the first bot detection work focusing on Polish Reddit. A two-stage annotation protocol yielded a comprehensive dataset, which has often been lacking in previous work on Reddit bot detection; the direct focus on this platform allowed us to exploit Reddit-specific conventions; targeting Polish communities prompted us to employ linguistic and cultural markers; SHAP values analysis revealed the importance and direction our features. Our models successfully incorporate explicit linguistic features in a multilingual setting and maintain strong performance in the 2025 testing. The quantitative analysis of the Polish bot ecosystem is another valuable contribution to both Polish-specific and multi-lingual research.

Limitations

Although we introduce several innovative features, their individual contributions warrant a more granular ablation analysis. Future work should evaluate the impact of feature groups, such as the linguistic features. The word counts of Polish, English, diacriticized, and potentially diacriticized words are, naturally, tightly connected to the overall verbosity of a comment, which is yet another characteristic that should be investigated more closely.

Unlike many prior studies — especially those incorporating Reddit data (Costa et al., 2015; Hurtado et al., 2019; Adel Alipour et al., 2023) — our temporal features exhibited relatively low predictive power. However, they may still be valuable in other settings. The temporal features we devised were arguably less rich and complex than those used in studies where temporal dynamics were a primary focus (Chavoshi et al., 2016; Mazza et al., 2019). Nonetheless, they may gain prominence in an author-level detection setting, where aggregate statistics could make behavioral regularities more salient.

Another potential limitation of our approach is the unweighted aggregation of model votes in the directional consensus report. Incorporating inputs from suboptimal models risks diluting the signal provided by top-performing classifiers.

The evident concept drift warrants a deeper exploration to isolate the underlying feature shifts and to determine the mechanisms that allow specific models to remain resilient despite the changing data distribution.

Although the two-stage annotation protocol enhances reliability, the external profile access introduces scraping dependencies and potential human biases.

Furthermore, dictionary-based features face inherent limitations when applied to morphologically rich languages like Polish. The rigidity of lexical matching may fail to account for complex inflection systems, diacritics, and orthographic variations.

References

Sanaz Adel Alipour, Rita Orji, and A. Zincir-Heywood. 2023. Behaviour and bot analysis on online social networks: Twitter, parler, and reddit. *International Journal of Technology and Human Interaction*, 19:1–19.

Andrea Bacciu, Massimo La Morgia, Alessandro Mei, Eugenio Nerio Nemmi, Valerio Neri, and Julinda Stefa. 2019. Bot and gender detection of twitter accounts using distortion and lsa. In *Conference and Labs of the Evaluation Forum*.

David Beskow and Kathleen Carley. 2018a. Bot conversations are different: Leveraging network metrics for bot detection in twitter. In *ASONAM*, pages 825–832.

David M. Beskow and Kathleen M. Carley. 2018b. Its all in a name: detecting and labeling bots by their name. *Computational and Mathematical Organization Theory*, 25:24 – 35.

Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2016. Debot: Twitter bot detection via warped correlation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 817–822.

Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2010. Who is tweeting on twitter: human, bot, or cyborg? In *Asia-Pacific Computer Systems Architecture Conference*.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Alceu Ferraz Costa, Yuto Yamaguchi, Agma J. M. Traina, Caetano Traina, and Christos Faloutsos. 2015. Rsc: Mining and modeling temporal activity in social media. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Stefano Cresci. 2020. A decade of social bot detection. *Communications of the ACM*, 63:72 – 83.

Robert Gorwa and Douglas Guilbeault. 2018. Unpacking the social media bot: A typology to guide research and policy. *Policy & Internet*.

Christian Grimme, Mike Preuss, Lena Adam, and Heike Trautmann. 2017. Social bots: Human-like by means of human control? *Big Data*, 5:279 – 293.

Sofia Hurtado, Poushali Ray, and Radu Marculescu. 2019. Bot detection in reddit political discussion. *Proceedings of the Fourth International Workshop on Social Sensing*.

Anna Kapuścińska. 2020. O emotikonach raz jeszcze – na przykładzie emotikonu "xd" w języku polskim. *Prace Językoznawcze*, 22(2):57–66.

Jürgen Knauth. 2019. Language-agnostic Twitter-bot detection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 550–558, Varna, Bulgaria. INCOMA Ltd.

Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *Information Sciences*, 467:312–322.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Michele Mazza, Stefano Cresci, Marco Avvenuti, Walter Quattrociocchi, and Maurizio Tesconi. 2019. Rtbust: Exploiting temporal patterns for botnet detection on twitter. *Proceedings of the 10th ACM Conference on Web Science*.

Lynnette Hui Xian Ng and Kathleen M. Carley. 2022. Botbuster: Multi-platform bot detection using a mixture of experts. *ArXiv*, abs/2207.13658.

Lynnette Hui Xian Ng and Kathleen M. Carley. 2024. Assembling a multi-platform ensemble social bot detector with applications to us 2020 elections. *Social Network Analysis and Mining*, 14:1–16.

Juan Pizarro. 2019. Using n-grams to detect bots on twitter: Notebook for pan at clef 2019. In *Working Notes Papers of the CLEF 2019 Evaluation Labs*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Adrian Rauchfleisch and Jonas Kaiser. 2020. The false positive problem of automatic bot detection in social science research. *PLoS ONE*, 15.

Stefan Stieglitz, Florian Brachten, Björn Ross, and Anna-Katharina Jung. 2017. Do social bots dream of electric sheep? a categorisation of social media bot accounts. *ArXiv*, abs/1710.04044.

Denis K. Stukal, Sergey Sanovich, Richard Bonneau, and Joshua A. Tucker. 2017. Detecting bots on russian political twitter. *Big Data*, 5:310 – 324.

Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. 2017. Online human-bot interactions: Detection, estimation, and characterization. In *International Conference on Web and Social Media*.

Kai-Cheng Yang, Onur Varol, Clayton A. Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2019a. Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies*.

Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2019b. Scalable and generalizable social bot detection through data selection. In *AAAI Conference on Artificial Intelligence*.

Kai-Cheng Yang, Onur Varol, Alexander C. Nwala, Mohsen Sayyadiharikandeh, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2023. Social bots: Detection and challenges. *Sociology, Social Policy and Education 2025*, abs/2312.17423.

Sarita Yardi, Daniel Romero, Grant Schoenebeck, and danah boyd. 2010. Detecting spam in a twitter network. *First Monday*, 15.

A Human-Bot Mixing Within a Comment

There was an author who had one comment in the dataset that appeared as though it could have been generated with the assistance of a conversational AI model, given its flawless grammar and punctuation, the use of an em dash (uncommon in informal social media discourse), and overall stylistic divergence from the rest of their textual content in their live profile. On the other hand, there were no other signs of automated activity and an age reference that appeared in this suspicious comment was consistent with the information shared elsewhere on the author's profile. Based on this consistency and the perceived human nature of the account, the comment was ultimately labeled human, though it was considered an edge case. The difficulty was partly due to the design of our labeling procedure, which had not accounted for potential human-bot mixing within a single comment; future studies may wish to explicitly address and define strategies for handling such ambiguous cases.

B Train Test Splits

Split	Subset	Total	Human	Bot	Bot (%)
CV Fold 0	Train	25842	10322	15520	60.06%
	Test	6461	2580	3881	60.07%
CV Fold 1	Train	25842	10321	15521	60.06%
	Test	6461	2581	3880	60.05%
CV Fold 2	Train	25842	10321	15521	60.06%
	Test	6461	2581	3880	60.05%
CV Fold 3	Train	25843	10322	15521	60.06%
	Test	6460	2580	3880	60.06%
CV Fold 4	Train	25843	10322	15521	60.06%
	Test	6460	2580	3880	60.06%
CV Total (Union)	Test	32303	12902	19401	60.06%
Temporal (2025)	Train	32303	12902	19401	60.06%
	Test	8484	4228	4256	50.17%

Table 2: Bot/Human counts for the CV and 2025 splits.

C Model Hyperparameters

Tab. 3 lists model hyperparameters.

D Comparing Feature Importance Across Models

One of the key challenges in the assessment of cross-model feature importance is the variability of the feature importance hierarchy. Not only is it different for each model type, it is also different in each run — e.g., the feature importance of the

Model	Hyperparameters
XGBoost	LR=0.099, Depth=7, N Est.=200, Subsample=0.8, Colsample=0.7
Random Forest	N Est.=220, Depth=None, Max Feat.=0.7, Min Split=4
Gradient Boost.	N Est.=300, LR=0.1, Depth=5, Max Feat.=None, Min Split=15, Subsample=1
SVM	C=4, Kernel=poly
Log. Reg. L1	C=100, Solver=saga, L1 Ratio=1
Log. Reg. L2	C=50, Solver=sag, L1 Ratio=0
N. Bayes Comp.	Alpha=0.5, Fit Prior=True
N. Bayes Mult.	Alpha=0.001, Fit Prior=True

Table 3: Hyperparameters of the CV and 2025 models.

cross-validated L1 Logistic Regression does not stay the same in the 2025 L1 Logistic Regression.

Figure 4 illustrates this divergence by plotting the cardinality of the union of top-N features across all models. The curve exhibits a steep initial slope, indicating high disagreement in the highest-ranked features. For example, at an arbitrary threshold of $N = 30$, the union expands to include 87 unique features — nearly three times the size of the threshold itself. This suggests that different models prioritize vastly different feature subsets. Conversely, the curve plateaus toward the upper limit (140 features), implying that while models disagree on what is important, they exhibit stronger consensus on the 'tail' — agreeing more on which features are irrelevant.

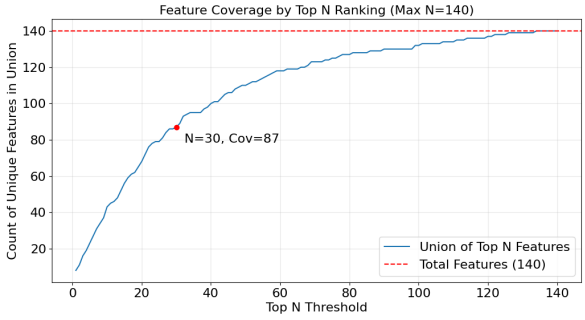

Figure 4: Feature coverage of top N features.

Furthermore, comparing importance is complicated by inconsistent feature importance scales. Models exhibit different dynamic ranges depending on their configuration — such as the C hyperparameter in Logistic Regression, which governs the penalization of coefficients and heavily influences the magnitude of the resulting importance scores. A partial solution to this is per-model normalization of SHAP values. Fig 5 illustrates normalized absolute mean feature importance of both the CV and 2025 models, while displaying their RRF ranking.

Feature Name	Signal	Avg Slope
bot_word_uppercase_in_name	High->Bot	0.86
platform_str_in_name	High->Bot	0.85
whitespace_entropy	High->Bot	0.83
platform_str	High->Bot	0.83
bot_word_capitalized_in_name	High->Bot	0.83
utility_str_in_name	High->Bot	0.82
capitalized_name	High->Bot	0.81
max_consecutive_carets	High->Bot	0.80
#cross_lvl_words_unique	High->Bot	0.79
beep_boop_str	High->Bot	0.79
max_consecutive_asterisks	High->Bot	0.79
m_a_bot_str	High->Bot	0.78
#carets	High->Bot	0.77
#exclamation_marks	High->Bot	0.77
i_am_a_bot_str	High->Bot	0.74
english_username_words_in_comments	High->Bot	0.73
emoticons_density	High->Bot	0.69
punctuation_entropy	High->Bot	0.69
utility_str	High->Bot	0.69
max_consecutive_exclamation_marks	High->Bot	0.68
#cross_lvl_words_total	High->Bot	0.68
#asterisks	High->Bot	0.68
#sentences	High->Bot	0.67
#m_dashes	High->Bot	0.65
#emojis	High->Bot	0.64
im_a_bot_str	High->Bot	0.63
#words_potential_diacritics	High->Bot	0.62
beep_boop_str_in_brackets	High->Bot	0.61
m_bot_str_in_brackets	High->Bot	0.59
#curly_brackets	High->Bot	0.48
priv_or_pv_word	High->Human	-0.38
#angle_brackets	High->Human	-0.53
xd_non_alphanum_density	High->Human	-0.60
#xd	High->Human	-0.60
xd_words_density	High->Human	-0.61
#polish_words	High->Human	-0.65
#rightwards_arrows	High->Human	-0.74
number_in_name	High->Human	-0.76
sentence_word_count_min	High->Human	-0.77
camel_case_name	High->Human	-0.83
polish_words_density	High->Human	-0.83
b_word_not_bot_in_name	High->Human	-0.85
sentences_capitalized_density	High->Human	-0.85
#digits_in_name	High->Human	-0.85
robot_in_name	High->Human	-0.89

Table 4: Perfect Consensus Features.

E Directional Consensus

Table 4 lists features with the perfect directional consensus, defined as the percentage of models that agree on the dominant sign of the slope (both the CV and 2025 models). The full report — including the ambiguous features — can be found in a separate CSV file.

Figure 5: Normalized feature importance of the CV and 2025 models, plus the RRF ranks.

F Feature List and Descriptions

We provide the full feature list and descriptions as a separate PDF file.