LSFES: A Linguistic Structure Feature Extraction System for Hate Speech and Offensive Language Classification

LSFES: A Linguistic Structure Feature Extraction System for Hate Speech And Offensive Language Classification

Waleed A.M. Alromaema
Dept. of Communications And Computer Networks Eng.
Politecnico di Torino
Torino, Italy
waleed.alromaema@studenti.polito.it

Claudio E. Casetti
Dept. of Control and Computer Eng.
Politecnico di Torino
Torino, Italy
claudio.casetti@polito.it

Aldo Franco Dragoni
Dept. of Information Eng.
Univ. Politecnica delle Marche
Ancona, Italy
a.f.dragoni@staff.univpm.it

***Abstract*—The rise in online interactions has increased hate speech, making its distinction from offensive language a key challenge. This paper introduces a novel methodology that combines structured-based and learning-based approaches to enhance classification. Traditional and deep learning-based features fail to retain contextual and grammatical meaning. By analyzing 1,700 sentences using constituency parse trees, we identified structural templates for hate speech. An Open Information Extraction System was developed to automate feature extraction using heuristic algorithms. We developed a Linguistic Structure Features Extraction System (LSFES). Our methodology shows a high precision in classification compared to the baseline.**

Keywords—hate speech, offensive language, feature extraction, open information extraction, natural language processing, natural language understanding, dependency parsing, machine learning, syntactic templates, Linguistic Structure Features Extraction System (LSFES).

I. INTRODUCTION

An exponential growth of online social interactions has led to a surge in hate speech [1], [2]. Detecting hate speech has become a critical challenge in digital communication, significantly complicating content moderation efforts [3], [4]. A fundamental challenge lies in distinguishing hate speech (which is often criminalized) from offensive language, as both share overlapping vocabulary but differ in intent and context [5], [6].

Traditional Natural Language Processing (NLP) methods frequently struggle with the complexity of contextual meanings, subtle linguistic nuances, and evolving hate speech patterns [7], [8]. Conventional feature extraction techniques, such as Bag of Words (BOW), n-grams, and TF-IDF, fail to retain contextual and grammatical relationships [7], [8]. Existing machine learning-based approaches often rely on handcrafted features or pre-trained word embeddings (e.g., Word2Vec), which may not fully capture the semantic and syntactic variations specific to hate speech [9], [10]. These limitations typically generate noisy feature vectors containing irrelevant terms, ultimately reducing classification precision [11].

This necessitates a novel methodology that integrates linguistic analysis to extract context-aware features for accurate hate speech detection [12], [13]. Online hate speech on social media platforms frequently escalates to real-world consequences: users exchange threats, online hostility transitions to physical confrontations, and verbal aggression progresses to criminal activities [1], [14]. When perpetrators become equipped with weapons, hate speech transforms into actionable violence that can engage hundreds of social media participants. Unregulated platforms accelerate social conflict by enabling rapid dissemination of inflammatory content, including hate-driven events, frightening imagery, and polarizing materials that manipulate community sentiments along political and religious affiliations [2], [15], [16].

This paper presents a novel technique for natural language feature extraction that enhances hate speech classification. By leveraging advanced linguistic feature engineering integrated with machine learning models, our method improves textual representation for more precise identification of harmful content.

II. RELATED WORK

In machine learning and pattern recognition, feature selection is crucial in classification tasks. For hate speech detection, various feature extraction approaches have been explored:

A. Surface-Level Features

The majority of existing works employ basic natural language processing (NLP) techniques: *Bag of Words (BoW)* and *n-grams* [1]–[7], [17] remain popular despite their simplicity, *character n-grams* [3], [11] address misspellings but increase dimensionality, and *TF-IDF* weighting [7] helps identify discriminative terms.

While these methods achieve reasonable performance, they fail to capture contextual relationships and grammatical structure [7], [8]. N-gram features extend BoW by generating word sequences of size N [1], [3], [5], with character n-grams

handling spelling variations [11]. However, these approaches lack dynamic context modeling [7].

B. Semantic and Neural Embeddings

Recent advances leverage distributed representations: *Word2Vec* and *fastText* [9] model semantic similarity, *document embeddings* [3] through vector averaging, and *Latent Dirichlet Allocation (LDA)* [8] for topic modeling.

These approaches improve context awareness through word clustering [7] and topic distributions [8], but lack explicit syntactic understanding [10].

C. Sentiment-Based Approaches

Given the correlation between hate speech and negative sentiment: Sentiment lexicons [3], [5], [6], [14] provide polarity scores, hybrid sentiment-classification pipelines [10], [17], [18], and emotion detection features [17].

However, sentiment analysis alone cannot distinguish hate speech from general negativity [5], despite high negative polarity correlation [5], [14].

D. Syntactic and Structural Features

Advanced linguistic analysis techniques include: *Part-of-Speech (POS) tagging* [5], [18], [19] for grammatical patterns, *dependency parsing* [10], [12] for long-range relationships, and *constituency trees* [20] for phrase structure analysis.

For example, in "he is a lower-class pig", dependency parsing reveals the critical *(pig, he)* relationship despite word distance [12]. However, POS tagging demonstrates questionable reliability in complex structures [5].

E. Template-Based Methods

Pattern-matching approaches using: hand-crafted templates (e.g., *I <intensity><userintent><hatetarget>*), regular expressions for specific hate patterns, and rule-based syntactic structures.

While achieving high precision (87-92%), these methods demonstrate limited recall (62-68%) [20] due to linguistic diversity in hate expressions. Manual relation extraction [10], [19] further limits scalability.

F. Research Gap

Existing approaches face key limitations: surface features ignore grammatical relationships, neural embeddings lack explicit syntactic modeling, and template methods cannot generalize to novel structures. Our *LSFES* addresses these gaps through automated syntactic-semantic pattern extraction from dependency parse trees. Unlike template-based approaches [20], LSFES employs heuristic algorithms to analyze all semantic relations and dependencies, generate generalized linguistic patterns, and maintain context sensitivity through parse tree analysis. This hybrid approach combines machine learning with linguistic rigor, enabling feature engineering that blends automated extraction with structural understanding [13], [16].

III. Research Objective

This paper proposes a hybrid framework combining structured linguistic analysis with machine learning (ML) to improve hate speech classification. The main objectives of our work are:

1) Develop a *Linguistic Structure Features Extraction System (LSFES)* through Open Information Extraction (OIE), creating context-rich features via syntactic-semantic parsing
2) Integrate linguistically-derived features with conventional ML classifiers to enhance detection accuracy
3) Establish an automated pipeline for linguistic feature extraction that encodes grammatical relationships and sentence structure

The system specifically addresses limitations in traditional methods by capturing structural patterns critical for hate speech context. For instance, hate speech often manifests through action-oriented verb structures (*e.g.*, "attack", "destroy"), while offensive language typically employs descriptive adjectives (*e.g.*, "stupid", "ugly"). LSFES encodes these distinctions through three core mechanisms: grammatical relationship mapping between distant sentence elements, verb-noun phrase interaction analysis, and structural template extraction from dependency parse trees.

This approach enables differentiation between superficially similar expressions through their underlying syntactic patterns, effectively addressing the critical challenge of context-dependent interpretation in hate speech detection.

IV. Methodology

We present a Linguistic Structure Features Extraction System (LSFES) to derive context-rich features via syntactic and semantic parsing. We used a comprehensive framework that utilizes a combination of feature extraction techniques and LSFES features. The methodology involves two workflows, as shown in Figure 1.

A. Structured-based System

The structured component employs LSFES rule-driven algorithms that craft rules, grammar, and linguistic knowledge to extract syntactical and semantic features. LSFES uses Heuristic algorithms to extract features like subject-object adjective-noun, subject-verb-object, and many syntactical and semantic relationships between message entities, as in Figure 1. It has three core modules: *Syntactic Parser*, which generates constituency and dependency parse trees; *Relation Extractor*, which identifies grammatical relationships using heuristic algorithms; and *Template Generator*, which creates linguistic patterns from frequent structures.

Key extracted features include:

$$\text{Template Score} = \frac{\sum(\text{Template})}{\text{Total clauses}} \tag{1}$$

where Template refers to syntactic or grammatical role patterns and or Part of speech POS, e.g. :

$$\text{SVO Score} = \frac{\sum(\text{Subject-Verb-Object triples})}{\text{Total clauses}} \tag{2}$$

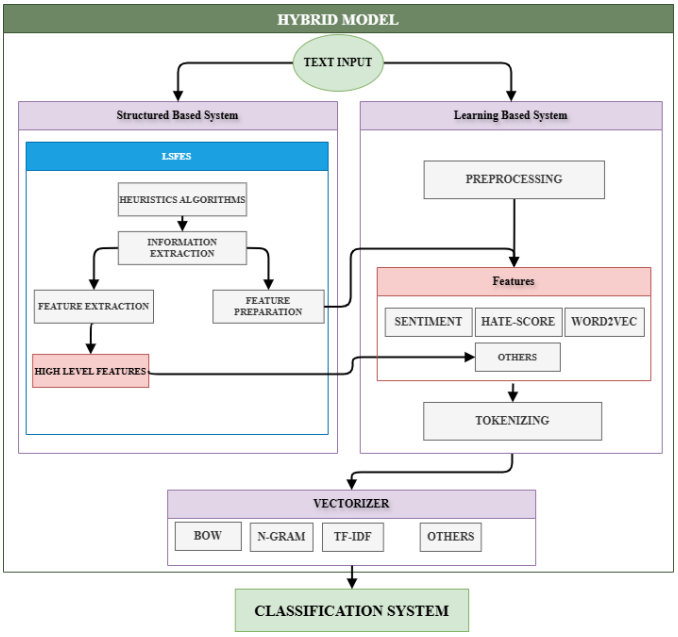


Fig. 1. Linguistic Structure Features Extraction System (LSFES) for hate speech and offensive language framework architecture

the top-rated templates were selected as templates that LSFES will extract from texts to generate features, and prepare a dataset in terms of meaningful features.

B. Learning-Based System

This section details the machine learning pipeline for hate speech classification, encompassing four key components: the annotated dataset, classification algorithms, feature extraction techniques, and integrated classification workflow.

1) Hate Speech Dataset: We have used a dataset that has been labeled by three to six experts [14] by Crowd Flower (CF) workers for annotation. They have been asked to consider the context in which Twitter users use it. They were instructed that the presence of a particular offensive word did not necessarily indicate that a tweet is hate speech. The results are 24,802 labeled tweets as summarized in Table I below:

TABLE I
ANNOTATION SUMMARY OF THE DATASET BY CF EXPERTS

Class	Number of Tweets	Percentage
Hate Speech	1,430	5.7%
Offensive Language	19,190	77.3%
Neither	4,163	16.7%
Unanimous	19	0.3%

2) Machine Learning Pipeline: The hybrid classification system combines linguistic feature extraction with machine learning through the integrated workflow illustrated in Figure 2. The pipeline operates through three core phases: **LSFES Feature Generation**, where the tool extracts linguistically meaningful features based on syntactic templates identified during our preliminary analysis (Section V); **Text Regeneration**, where LSFES reconstructs natural language expressions preserving semantic relationships while optimizing for feature extraction; and **Feature Enhancement**, where regenerated text serves as enriched input for downstream feature extraction processes.

As detailed in Section V, LSFES performs dual functionality - it both analyzes existing text structures and generates optimized textual representations that improve subsequent machine learning performance. This bidirectional capability allows the system to maintain original semantic meaning, enhance syntactic relationships, and improve feature discriminability.

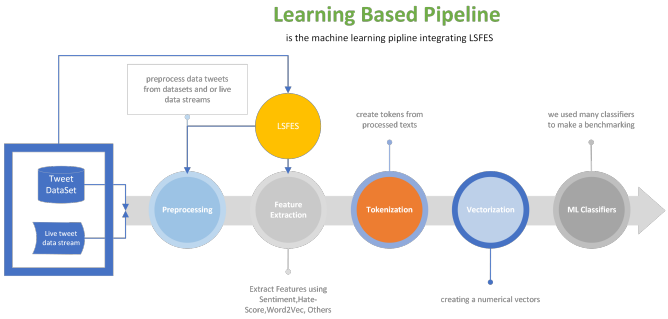


Fig. 2. End-to-end machine learning pipeline integrating Linguistic Structure Features Extraction System (LSFES) features

V. LSFES (LINGUISTIC STRUCTURE FEATURES EXTRACTION SYSTEM)

LSFES is an open information extraction (OIE) tool designed for the preparation and extraction of natural language linguistic features. It was developed based on linguistic principles, phenomena, rules, and assumptions. The tool was developed through the analysis of hate speech and offensive language, and has the potential to be generalized to cover broader natural language linguistic feature extraction.

A. Background and Vision

Machine learning performance for hate speech detection depends critically on both data quality and feature representation. However, natural language's inherent complexity—particularly in social media—poses significant key challenges, including contextual ambiguity blurring hate speech/offensive language boundaries, conventional features failing to capture semantic relationships, and expert annotation difficulties due to overlapping terminology.

Our methodology addresses these challenges through linguistic structure analysis, proposing that *precise identification of term relationships and their machine-interpretable representation can optimize classification accuracy*. The challenges include contextual ambiguity, blurring hate speech/offensive language boundaries, conventional features failing to capture semantic relationships, and expert annotation proving difficult due to overlapping terminology.

B. Linguistic Pattern Extraction

Our analysis of 1,700 hate-containing tweets revealed consistent syntactic patterns through constituency and dependency parsing. Figure 3 illustrates three characteristic configurations of offensive phrases as `make fa*got` [1]:

1) Pattern Discovery Methodology: The extraction process involved:

1) Parse tree generation using Stanford CoreNLP [2]
2) Hate term localization in syntactic constituents
3) Relationship analysis between hate terms and targets
4) Frequency analysis of structural patterns

This revealed that hate expressions predominantly follow specific grammatical templates:

$$\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_n\} \tag{3}$$

where each template τ_i represents a verified hate speech pattern.

2) Template Taxonomy: The identified patterns form three principal categories as shown in Table II: canonical patterns for basic hate actions, descriptive patterns for pejorative labeling, and complex patterns for multi-class combinations. These templates capture essential grammatical relationships that distinguish hate speech from offensive language.

TABLE II
LINGUISTIC PATTERNS FOR HATE SPEECH STRUCTURES

Type	Pattern	Description
Canonical	`<Subj><Verb><Obj>`	Basic hate actions
	`<Subj><Adv><Verb><Obj>`	Intensified actions
Descriptive	`<Adj><Noun>`	Pejorative labeling
	`<Subj><Adj><Obj>`	Attribute assignment
Complex	`<mix Adj V N>`	Multi-class combinations
	`Nested structures`	Phrasal constructions

Figure 4 demonstrates how these templates generate linguistically informed features, contrasting with superficial n-gram approaches that lose structural relationships.

3) Advantages Over Traditional Features: The template approach addresses key limitations of **Bag-of-Words**, which lacks grammatical relationships; **TF-IDF**, which ignores term positioning; and **Word Embeddings**, which obscure syntactic roles.

C. Template Extraction Architecture

The LSFES design implements a pipeline for structured feature extraction through syntactic analysis of dependency trees. As shown in Figure 1, The extraction algorithms summarized in Table III enable automated identification of key grammatical relationships in dependency parses. These heuristics target specific dependency types to systematically extract subject-verb-object triples and modifier relationships. It shows six specialized extraction algorithms that operate on Stanford Dependency parses:

[1]We have replaced letters of offensive words with '*' to avoid writing them plainly in a scientific paper

[2]http://nlp.stanford.edu/software/corenlp.shtml

TABLE III
EXTRACTION ALGORITHMS AND THEIR FUNCTIONS

Algorithm	Extraction Pattern
Subject Extraction	Identifies hate sources via `nsubj` relations
Verb Extraction	Captures action terms through `root` and `acl`
Object Extraction	Locates targets using `dobj` and `nmod`
Adverbial Extraction	Marks intensifiers via `advmod`
Adjectival Extraction	Finds descriptors through `amod`
Nominal Extraction	Processes noun phrases via `compound`

D. Feature Generation Analysis

1) Case Study: Predicate Analysis: Figure 5 illustrates feature extraction from the sentence *"Anna kills a wild wolf and takes it away"* for hate speech and offensive language detection. The dependency tree reveals key predicate-argument structures. Two meaningful dependency paths are extracted: the first includes *Anna kills*, *kills wolf*, and *wild wolf*, highlighting potentially violent expressions; the second includes *Anna takes*, *Anna takes it*, and *takes it*, reflecting control-related language. Such syntactic features help identify harmful or aggressive content beyond the presence of explicit keywords.

The system generates linguistically motivated features as shown in Table IV, which presents the relational composition and weighting components. :

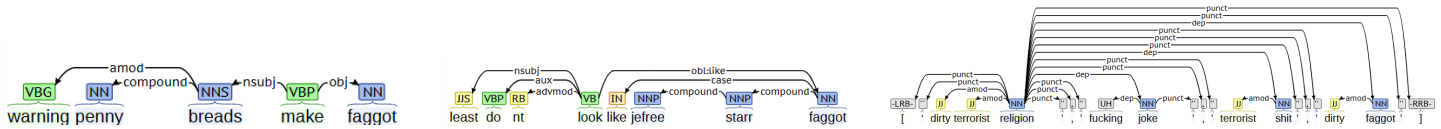
TABLE IV
RELATIONAL DECOMPOSITION AND WEIGHTING COMPONENTS

Component	Details
Phrasal Decomposition	`Anna kills`, `kills wolf`, `wild wolf`, `Anna takes`, and `takes it`.
Relational Weighting	$w(r) = \frac{\text{depth}(r)}{\text{max\_depth}} \times \text{TF}(r)$ where r denotes syntactic relationships (e.g., `<verb-object>` or `<adjective-noun>`), depth(r) represents tree distance between terms, max_depth indicates the deepest level in parse tree, TF(r) is document frequency of r, and $w(r)$ is the normalized weight.
Example Calculation	*Sentence:* "Anna kills a wild wolf and takes it away" • `kills` → `wolf` (verb-object): – depth(r) = 2, max_depth = 3 – TF(r) = 2 ⇒ $w(r) \approx 1.33$ • `wild` → `wolf` (adjective-noun): – depth(r) = 1, max_depth = 3 – TF(r) = 1 – $w(r) \approx 0.33$
Semantic Network Construction	(See Fig. 6)

2) Feature Characteristics: The extracted features exhibit three key properties, as detailed in Table V, which describes the feature representation properties and their implementation methodology:

VI. RESULTS AND DISCUSSION

The model was evaluated through multiple runs, and the performance metrics of the classifiers are summarized in Table VI. The probabilistic classifiers, namely Multinomial Naïve Bayes and Logistic Regression, achieved the highest



(a) Verbal phrase `make fa*got`

(b) Noun modification `jefree starr fa*got`

(c) Mixed patterns `d*rty terrorist religion fu*king joke, terrorist sh*t`

Fig. 3. Syntactic patterns in hate speech: (a) Verbal Phrase (VP)-contained hate terms, (b) Noun Phrase (NP)-based modification, (c) Complex adjective-noun combinations.

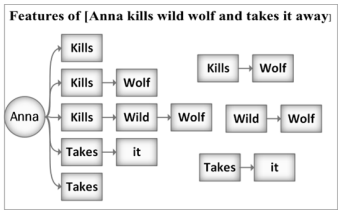


Fig. 4. Feature generation comparison: (Top) Conventional n-grams vs (Bottom) Linguistic Structure Features Extraction System (LSFES) template-derived features showing preserved syntactic relationships

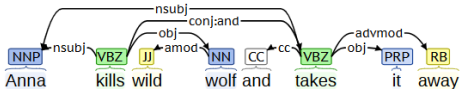


Fig. 5. Dependency tree illustrating meaningful features extracted from two dependency tree paths. The first path includes the phrases: *Anna kills*, *kills wolf*, and *wild wolf*. The second path includes *Anna takes*, *Anna takes it*, and *takes it*. These features represent key syntactic and semantic relationships captured from the dependency structure of the sentences.

average scores along with strong classification recall. Among these, Multinomial Naïve Bayes exhibited the best classification performance. Linear classifiers, including SVM, Linear SVC, and Perceptron, also demonstrated competitive results. In contrast, tree-based classifiers such as Random Forest and SGD Classifier showed comparatively lower performance, likely due to their hard decision boundaries, which negatively impacted classification accuracy. Notably, Random Forest exhibited a lower recall for hate speech detection.

The Multinomial Naïve Bayes classifier emerged as the best-performing model. The confusion matrix (left panel of Figure 7) reveals that approximately 30% of the *Hate* class

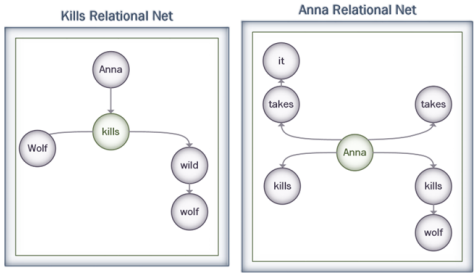


Fig. 6. Semantic network showing term centrality in feature space

TABLE V
FEATURE REPRESENTATION PROPERTIES AND IMPLEMENTATION METHODOLOGY

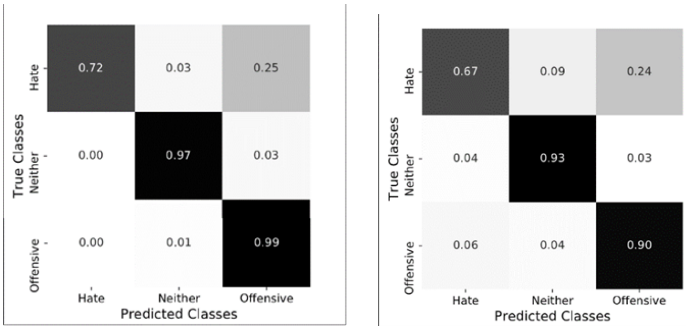
Property	Description
Compositionality	Preserves phrase structure in feature representation.
Interpretability	Maintains a human-readable form of linguistic features.
Discriminability	Enhances class separation through: $$\mathcal{D}(f) = \frac{\lvert\mu_1(f) - \mu_2(f)\rvert}{\sigma_1(f) + \sigma_2(f)}$$ where f denotes linguistic features (e.g., verb-object pairs), μ_1 and μ_2 represent mean frequency in hate and non-hate classes respectively, σ_1 and σ_2 indicate standard deviations per class, and $\mathcal{D}(f)$ is the discriminability score. **Example Calculations**: • `kills → wolf`: – $\mu_1 = 0.75$, $\mu_2 = 0.10$ – $\sigma_1 = 0.15$, $\sigma_2 = 0.05$ – $\mathcal{D}(f) = 3.25$ • `takes → it`: – $\mu_1 = 0.20$, $\mu_2 = 0.18$ – $\mathcal{D}(f) \approx 0.13$
Implementation Steps	1) **Feature Extraction**: • Extract verb-object pairs (e.g., `"kills wolf"`) • Tag with parse tree depth (e.g., `depth=2`) 2) **Statistical Profiling**: $$\mu_1(f) = \text{Mean in hate class}$$ $$\sigma_1(f) = \text{Std. dev. in hate class}$$ 3) **Weighted Feature Selection**: $$w(f) = \underbrace{\frac{\text{depth}(f)}{\text{max\_depth}}}_{\text{Syntactic weight}} \times \mathcal{D}(f)$$ • f: Linguistic feature • depth(f): Parse tree depth • $\mathcal{D}(f)$: Discriminability score **Example**: • `"kills → wolf"`: – depth $= 2$, max_depth $= 3$ – $\mathcal{D}(f) = 3.25$ – $w(f) \approx 2.18$

samples were misclassified, with 25% incorrectly labeled as *Offensive*. This misclassification likely stems from the inherent overlap between hate speech and offensive language, a challenge also encountered during manual annotation. Notably, hate speech constitutes only 5.7% (1,430 out of 25,000 tweets) of the dataset, which may further complicate classification.

The Perceptron, a neural network-based classifier, achieved the second-best performance, with an average precision of 0.91, recall of 0.88, and F1-score of 0.89. Its confusion

TABLE VI
CLASSIFIER PERFORMANCE WITH CONFIDENCE INTERVALS AND AVERAGE SCORES

Classifier	Confidence Interval	Average Score
SVM	[0.9540, 0.94876, 0.9495, 0.9455, 0.9491]	0.9493
Logistic Regression	[0.9572, 0.94675, 0.9487, 0.9491, 0.9487]	0.9500
Multinomial NB	[0.97095, 0.96853, 0.97176, 0.9665, 0.9657]	0.9686
Random Forest	[0.89269, 0.88584, 0.9027, 0.8777, 0.9048]	0.8927
SGD Classifier	[0.91004, 0.90197, 0.9048, 0.88180, 0.89552]	0.8988
Linear SVC	[0.9556, 0.94392, 0.9455, 0.9447, 0.94634]	0.9472
Perceptron	[0.8624, 0.87051, 0.8773, 0.77006, 0.79870]	0.8357
Decision Tree	[0.9148, 0.9124, 0.9015, 0.8967, 0.90802]	0.9066



(a) Multinomial Naïve Bayes (b) Perceptron

Fig. 7. Confusion matrices for (a) Multinomial Naïve Bayes and (b) Perceptron classifiers

matrix (right panel of Figure 7) indicates that 24% of hate speech tweets were misclassified as offensive, reinforcing the observed semantic overlap between the two categories.

VII. CONCLUSION

Our LSFES framework demonstrates that syntactic-semantic pattern analysis significantly improves hate speech detection accuracy. The hybrid approach achieves an average of 94-95% precision, outperforming traditional methods. Future work will explore deep learning integration and cross-lingual applications.

VIII. ETHICAL IMPLICATIONS AND RESPONSIBLE DEPLOYMENT

Key considerations include: bias amplification risks from data imbalances (only 5.7% hate speech in the dataset), Inherited cultural biases disproportionately flag marginalized groups, misclassification of reclaimed slurs and sarcasm, censorship risks for legitimate discourse, human-AI collaboration requirements with transparent appeals, continuous bias audits across demographic groups, adversarial retraining with benign examples, contextual safeguards incorporating user history and platform norms, transparency through metrics disclosure and DM opt-outs, and purpose limitation to harm prevention with data anonymization. These measures ensure responsible implementation while mitigating ethical risks.

REFERENCES

[1] D. Hovy and W. Zeerak, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL Student Research Workshop*, San Diego, California, USA, 2016, pp. 88–93.

[2] C. V. Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. D. Pauw, and V. Hoste, "Detection and fine-grained classification of cyberbullying events," in *Proceedings of Recent Advances in Natural Language Processing*, Hissar, Bulgaria, 2015, pp. 672–680.

[3] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th International Conference on World Wide Web*, Geneva, Switzerland, 2016, pp. 145–153.

[4] H. Hosseinmardi, S. A. Mattson, R. Ibn Rafiq, R. Han, Q. Lv, and S. Mishra, "Detection of cyberbullying incidents on the instagram social network," *CoRR*, vol. abs/1503.03909, 2015.

[5] P. Burnap and M. L. Williams, "Identifying cyber hate on twitter across multiple protected characteristics," *EPJ Data Science*, vol. 5, no. 1, pp. 1–15, 2016.

[6] J.-M. Xu, K.-S. Jun, X. Zhu, and A. Bellmore, "Learning from bullying traces in social media," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montréal, Canada, 2012, pp. 656–666.

[7] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proceedings of the Second Workshop on Language in Social Media*, Stroudsburg, PA, USA, 2012, pp. 19–26.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of Workshop at the International Conference on Learning Representations (ICLR)*, Scottsdale, AZ, USA, 2013.

[10] N. D. Gitari, Z. Zuping, H. Damien, and J. Long, "A lexicon-based approach for hate speech detection," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 4, pp. 215–230, 2015.

[11] Y. Mehdad and J. Tetreault, "Do characters abuse more than words?" in *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Los Angeles, CA, USA, 2016, pp. 299–303.

[12] Y. Chen, "Detecting offensive language in social medias for protection of adolescent," Ph.D. dissertation, The Pennsylvania State University, State College, PA, USA, 2011.

[13] O. Etzioni *et al.*, "Open information extraction: The second generation," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, Catalonia, Spain, 2011, pp. 3–10.

[14] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," arXiv preprint arXiv:1703.04009, 2017.

[15] V. Singh and B. Kumar, "Feature extraction techniques for handwritten," *International Journal of Soft Computing and Engineering*, vol. 238, 2013.

[16] L. Del Corro and R. Gemulla, "Clausie: Clause-based open information extraction," in *Proceedings of the International World Wide Web Conference*, 2013.

[17] S. O. Sood, E. F. Churchill, and J. Antin, "Automatic identification of personal insults on social news sites," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 2, pp. 270–285, 2012.

[18] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," *The Social Mobile Web*, vol. 11, no. 2, 2011.

[19] E. Greevy and A. F. Smeaton, "Classifying racist texts using a support vector machine," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2004, pp. 468–469.

[20] M. Mondal, L. A. Silva, and F. Benevenuto, "A measurement study of hate speech in social media," *arXiv preprint arXiv:1703.04009*, 2017.