

SAVEETHA SCHOOL OF ENGINEERING

SIMATS, CHENNAI - 602105



CSA1353-THEORY OF COMPUTATION FOR CHOMSKY NORMAL FORM

SPAM CLASSIFICATION

Guided BY,
Dr . K . Thinakaran
(*course faculty*)
Theory Of Computation
SSE,SIMATS

Presented by:-
1. K.Dinesh (192210140)
2. J . Harsha Vardhan Reddy (192210422)
3. M . Dinesh Reddy(192211451)
CSE DEPARTMENT
SSE,SIMATS



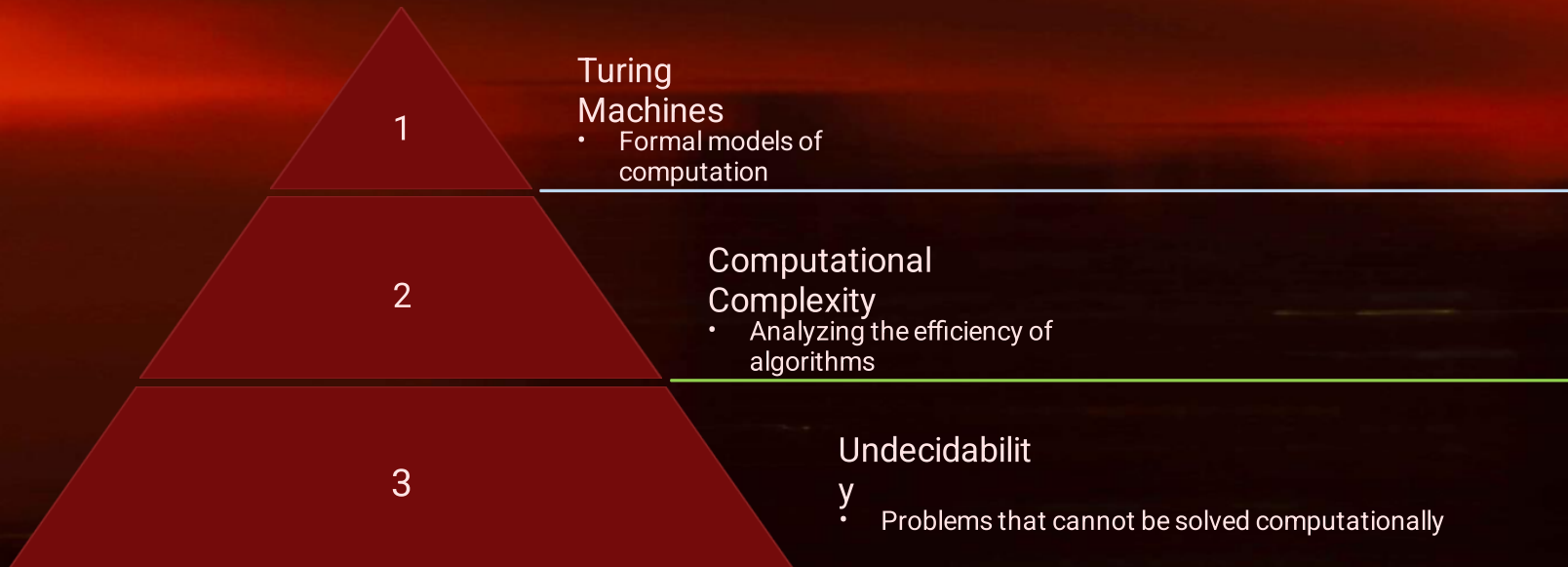
Wondershare PDFElement

Introduction to Spam Classification

- Spam Classification is the process of identifying and filtering out unwanted or unsolicited messages, commonly referred to as spam, from legitimate communication. This is particularly important for email systems, social media platforms, and online forums to ensure the quality and security of information flow.
- Spam refers to unsolicited, irrelevant, or inappropriate messages sent over the internet, typically to a large number of users, for the purposes of advertising, phishing, spreading malware, etc.
- Spam classification is a crucial task in modern email and communication systems. By accurately identifying and filtering out unwanted, unsolicited messages, users can maintain a clean and focused inbox, improving productivity and digital well-being.



Fundamentals of Theory of Computing



- Theory of computing is the foundational discipline that underpins our understanding of what can and cannot be computed by machines. Key concepts include Turing machines as models of computation, the analysis of computational complexity, and the study of undecidable problems that lie beyond the reach of algorithms.



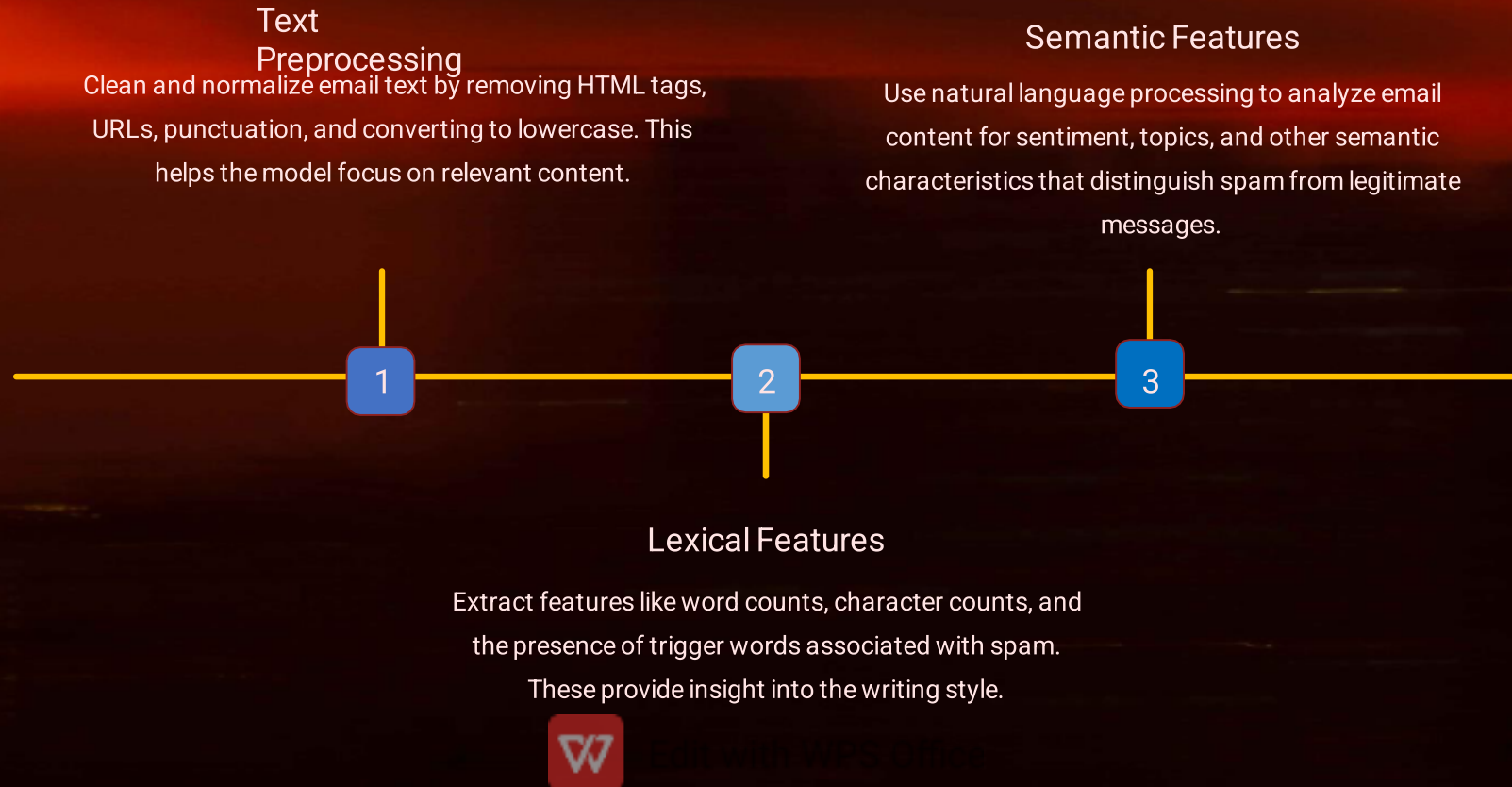
The Spam Classification Problem



- The spam classification problem involves developing intelligent systems to automatically sort through the vast influx of emails and separate legitimate messages from unwanted, malicious spam. This is a crucial task to safeguard users' inboxes, prevent financial losses, and maintain productivity in the face of increasingly sophisticated spam tactics.



Feature Engineering for Spam Classification



Computational Complexity of Spam Classification

1

Polynomial-Time Algorithms

- Many spam classification algorithms, such as naive Bayes and logistic regression, are known to have polynomial-time complexity, making them efficient for practical use.

2

NP-Hardness Challenges

- However, certain variants of the spam classification problem, such as those involving adversarial attacks, have been shown to be NP-hard, posing computational challenges.

3

Tradeoffs and Approximations

- Researchers often explore tradeoffs between classification accuracy and computational complexity, as well as approximate algorithms to tackle the NP-hard aspects of spam classification.



CODE

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
typedef struct {
    char text[100];
    int is_spam;
} Message;
int classify_message(Message message) {
    char* spam_keywords[] = {"free", "money", "click", "here", "now", "limited", "time", "offer"};
    int num_spam_keywords = sizeof(spam_keywords) / sizeof(char*);
    for (int i = 0; i < num_spam_keywords; i++) {
        if (strstr(message.text, spam_keywords[i]) != NULL) {
            return 1;
        }
    }
    return 0;
}
int main() {
    Message training_messages[] = {
        {"Free money! Click here!", 1},
        {"Make money fast!", 1},
        {"Hello, how are you?", 0},
        {"I love you!", 0},
        {"Limited time offer! Don't miss out!", 1},
        {"This is a legitimate email.", 0}
    };
    int num_training_messages = sizeof(training_messages) / sizeof(Message);
    Message test_messages[] = {
        {"Make money fast! Click here!", -1},
        {"Hello, how are you?", -1},
        {"Free gift! Limited time only!", -1},
        {"This is a spam email.", -1},
        {"I love you!", -1}
    };
    int num_test_messages = sizeof(test_messages) / sizeof(Message);
    for (int i = 0; i < num_test_messages; i++) {
        int classification = classify_message(test_messages[i]);
        printf("Message: %s\n", test_messages[i].text);
        printf("Classification: %s\n\n", classification == 1 ? "Spam" : "Not spam");
    }
    return 0;
}
```



Edit with WPS Office

Programiz

C Online Compiler

Programiz

PRO

Premium Coding Courses

by Programiz

Learn More

Programiz PRO

>

main.c

Run

Output

Clear

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 typedef struct {
5     char text[100];
6     int is_spam;
7 } Message;
8 int classify_message(Message message) {
9     char* spam_keywords[] = {"free", "money", "click", "here", "now", "limited",
10                             "time", "offer"};
11     int num_spam_keywords = sizeof(spam_keywords) / sizeof(char*);
12     for (int i = 0; i < num_spam_keywords; i++) {
13         if (strstr(message.text, spam_keywords[i]) != NULL) {
14             return 1;
15         }
16     }
17     return 0;
18 }
19 int main() {
20     Message training_messages[] = {
21         {"free money! click here!", 1},
22         {"Make money fast!", 1},
23         {"Hello, how are you?", 0},
24         {"I love you!", 0},
25         {"limited time offer! Don't miss out!", 1},
26     };
27     int num_training_messages = sizeof(training_messages) / sizeof(Message);
28     Message test_messages[] = {
29         {"Make money fast! Click here!", -1},
30         {"Hello, how are you?", 1},
31         {"free gift! limited time only!", 1},
32         {"This is a spam email.", -1},
33         {"I love you!", -1}
34     };
35     int num_test_messages = sizeof(test_messages) / sizeof(Message);
36     for (int i = 0; i < num_test_messages; i++) {
37         int classification = classify_message(test_messages[i]);
38         printf("Message: %s\n", test_messages[i].text);
39         printf("Classification: %s\n", classification == 1 ? "Spam" : "Not
40             spam");
41     }
42     return 0;
43 }
```

```
/tmp/onM5rNPNGs.o
Message: Make money fast! Click here!
Classification: Spam

Message: Hello, how are you?
Classification: Not spam

Message: Free gift! Limited time only!
Classification: Spam

Message: This is a spam email.
Classification: Not spam

Message: I love you!
Classification: Not spam

--- Code Execution Successful ---

Classification: Spam
Message: this is a spam email.
Classification: Not spam
Message: I love you!
Classification: Not spam

--- Code Execution Successful ---
```

WPS Office

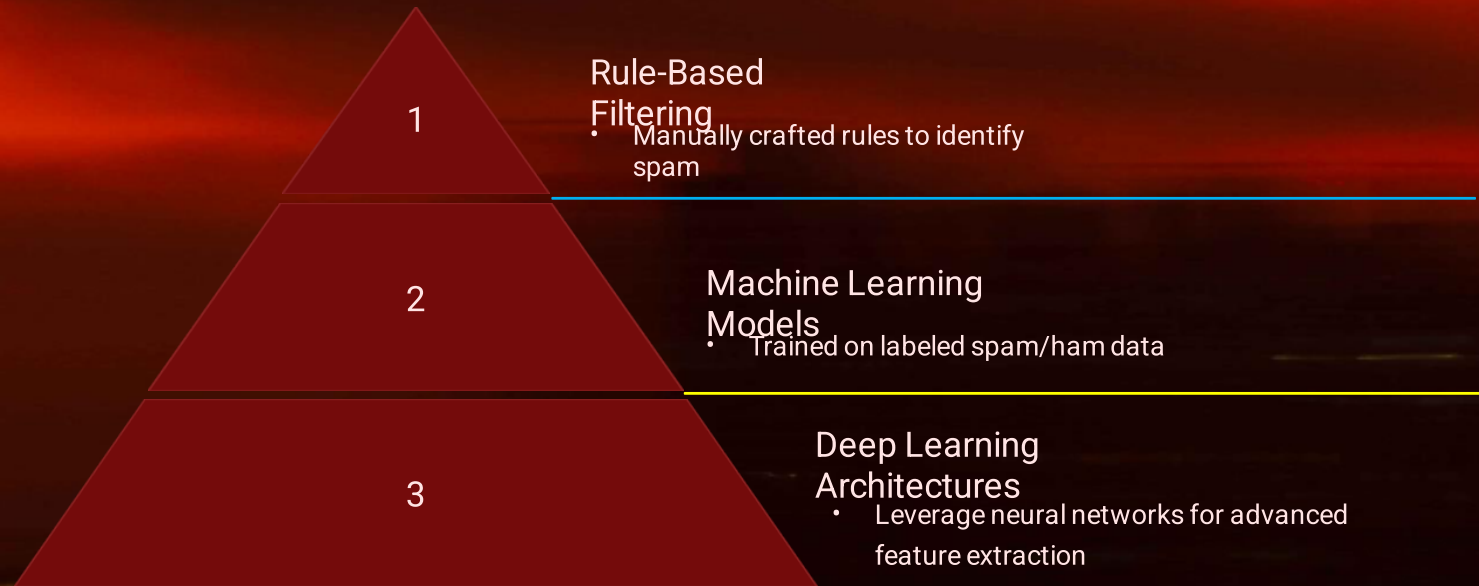
Edit with WPS Office

BSI midcap -1.02%

Search

12:57 24-06-2024

Algorithmic Approaches to Spam Classification



- Spam classification has been approached using a variety of algorithmic techniques. Traditional rule-based filters rely on manually curated heuristics to identify spam messages. More advanced approaches leverage machine learning models, including linear classifiers, decision trees, and ensemble methods, trained on large datasets of labeled spam and ham emails. The cutting edge of spam filtering utilizes deep neural networks that can automatically extract rich features from email content and metadata.

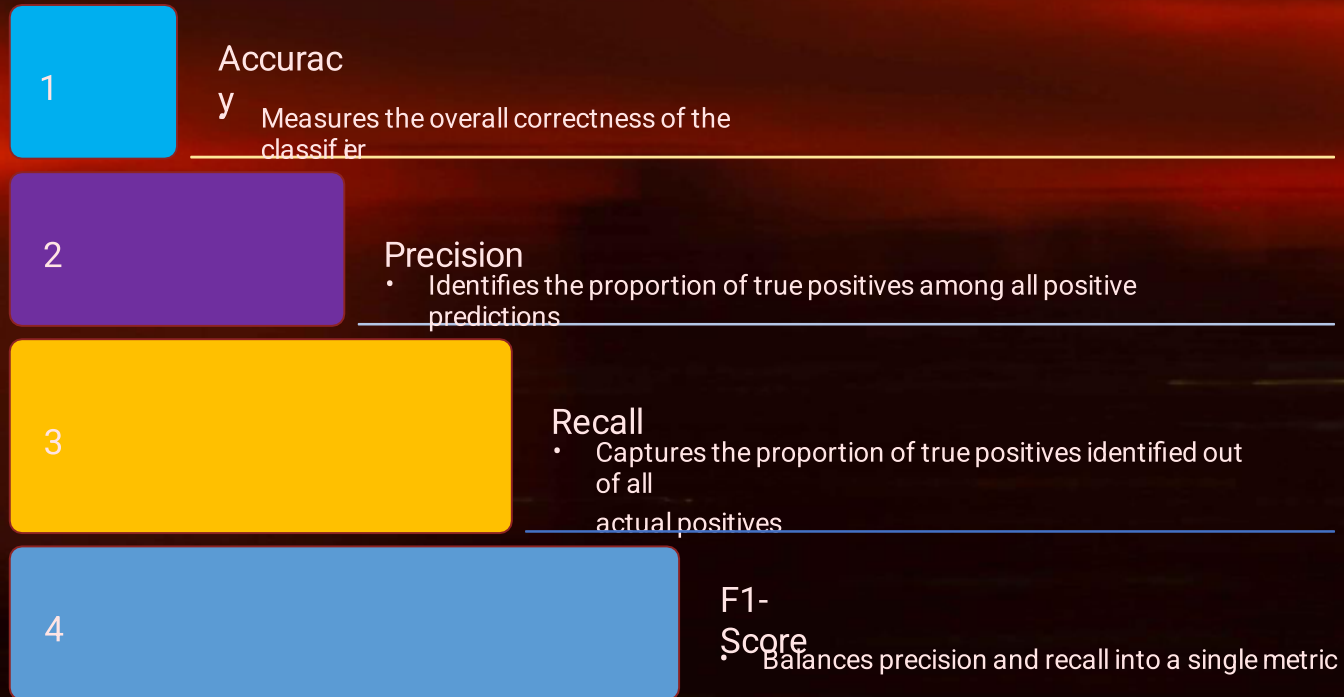


Spam Classification using Naive bayes Algorithm

- Naive Bayes is a probabilistic classifier based on Bayes' Theorem with the assumption of independence between features. It is particularly effective for text classification tasks like spam detection.
- Naive Bayes is a probabilistic classifier based on Bayes' Theorem with the "naive" assumption that the features are conditionally independent given the class label. It is particularly popular for text classification tasks such as spam detection. Here is a step-by-step explanation of how Naive Bayes can be used for spam classification, including the relevant formulae.
- Why Naive Bayes?
Simplicity: Easy to implement and understand. Efficiency: Requires less computational resources. Effectiveness: Works well with text data, especially when the feature set is large.
- Formulae:
 - $P(C/X) = P(X/C) \cdot P(C) / P(X)$
 - $P(C) = \text{Number of documents in class C} / \text{Total number of documents}$
 - $P(X) = \text{Number of variables} / \text{total classes}$



Evaluation Metrics and Benchmarking



- Evaluating the performance of spam classification models is crucial to ensure they are effective and reliable. Key metrics include accuracy, precision, recall, and the F1-score, which provide insights into the classifier's ability to correctly identify spam and non-spam messages. Benchmarking against industry standards and cross-validating results help validate the model's generalization capabilities.



Limitations and Challenges



1

Evolving Spam Tactics

- Spammers are constantly devising new techniques to bypass spam filters, making it an ongoing battle to stay ahead of their tactics.

2

Data Imbalance

- Spam classification often suffers from highly imbalanced datasets, with legitimate emails vastly outnumbering spam messages, making it challenging to train effective models.

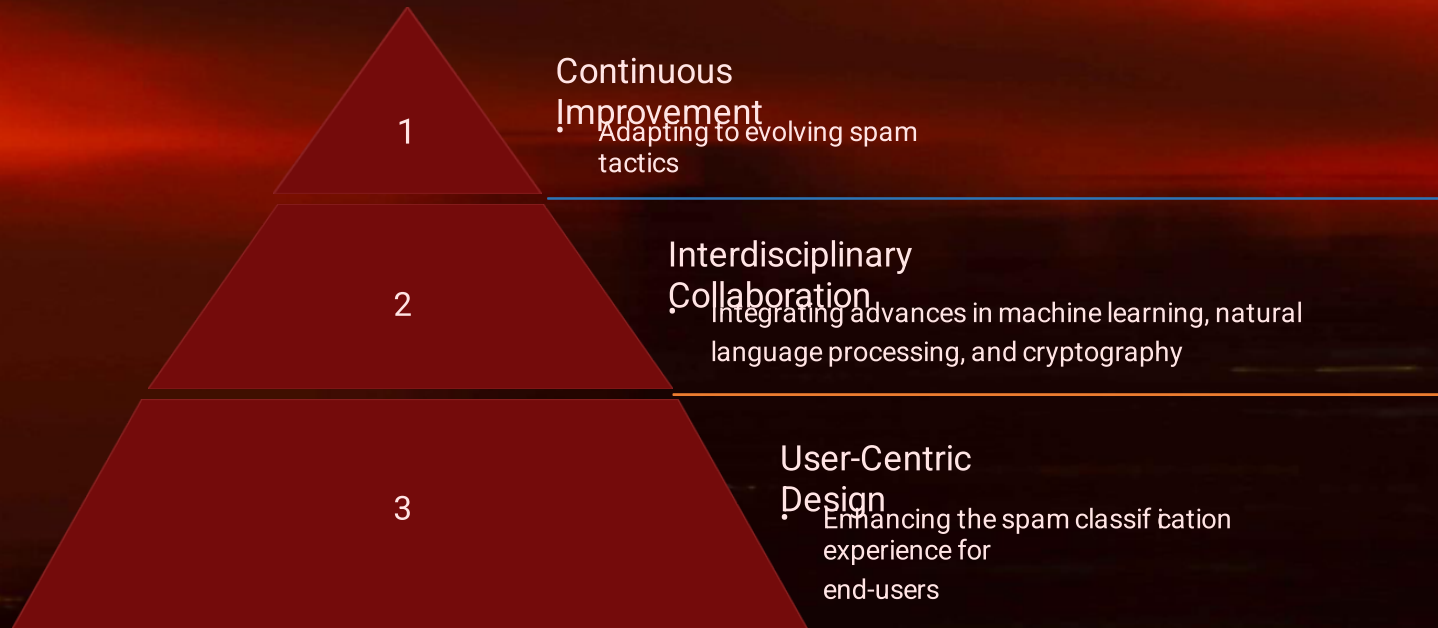
3

Context Dependency

- The meaning and intent of emails can be highly context-dependent, making it difficult for algorithms to accurately distinguish between spam and legitimate messages.



Conclusion and Future Directions



- In conclusion, spam classification remains an active area of research and development, requiring continuous innovation to stay ahead of evolving spam tactics. Future advancements will likely involve deeper interdisciplinary collaboration, leveraging the latest breakthroughs in machine learning, natural language processing, and cryptography. Ultimately, the goal is to deliver a seamless, user-centric spam classification experience that protects individuals and organizations from the ever-growing threat of unwanted and malicious messages.



**THANK
YOU**



Edit with WPS Office