SENTENCE AUTOCOMPLETE

Akash .T - 192210307
Naga Malli karjun. L - 192211444
S K Shanwaz- 192110359

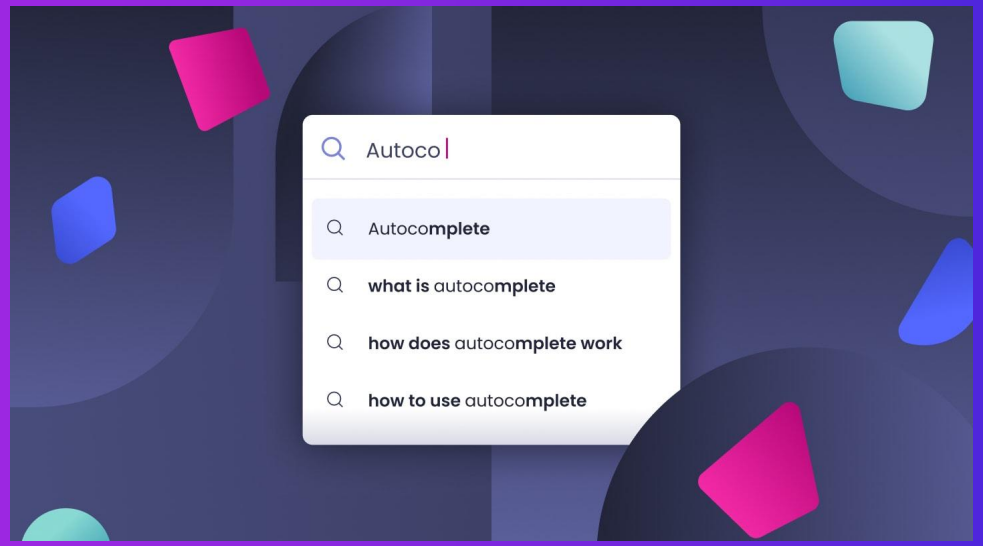
Table Of Contents:

Introduction

- The Sentence Auto complete project aims to improve typing efficiency and user experience by predicting and suggesting the next words or phrases in a sentence.
- Utilizing machine learning and natural language processing (NLP), the system understands context to provide accurate predictions.
- This technology can be used in various applications, such as email composition, coding, messaging apps, and other text input interfaces, to enhance productivity and ease of use.
- By reducing the amount of typing required, Sentence Auto complete streamlines the writing process and ensures a smoother, more efficient user experience.

Objectives

- **Predict Next Words**: Develop a system that can predict and suggest the next words or phrases in a sentence.
- **Improve Typing Efficiency**: Reduce the amount of typing required by users.
- **Enhance User Experience**: Provide accurate and contextually relevant suggestions.
- **Support Multiple Languages**: Extend the autocomplete functionality to multiple languages

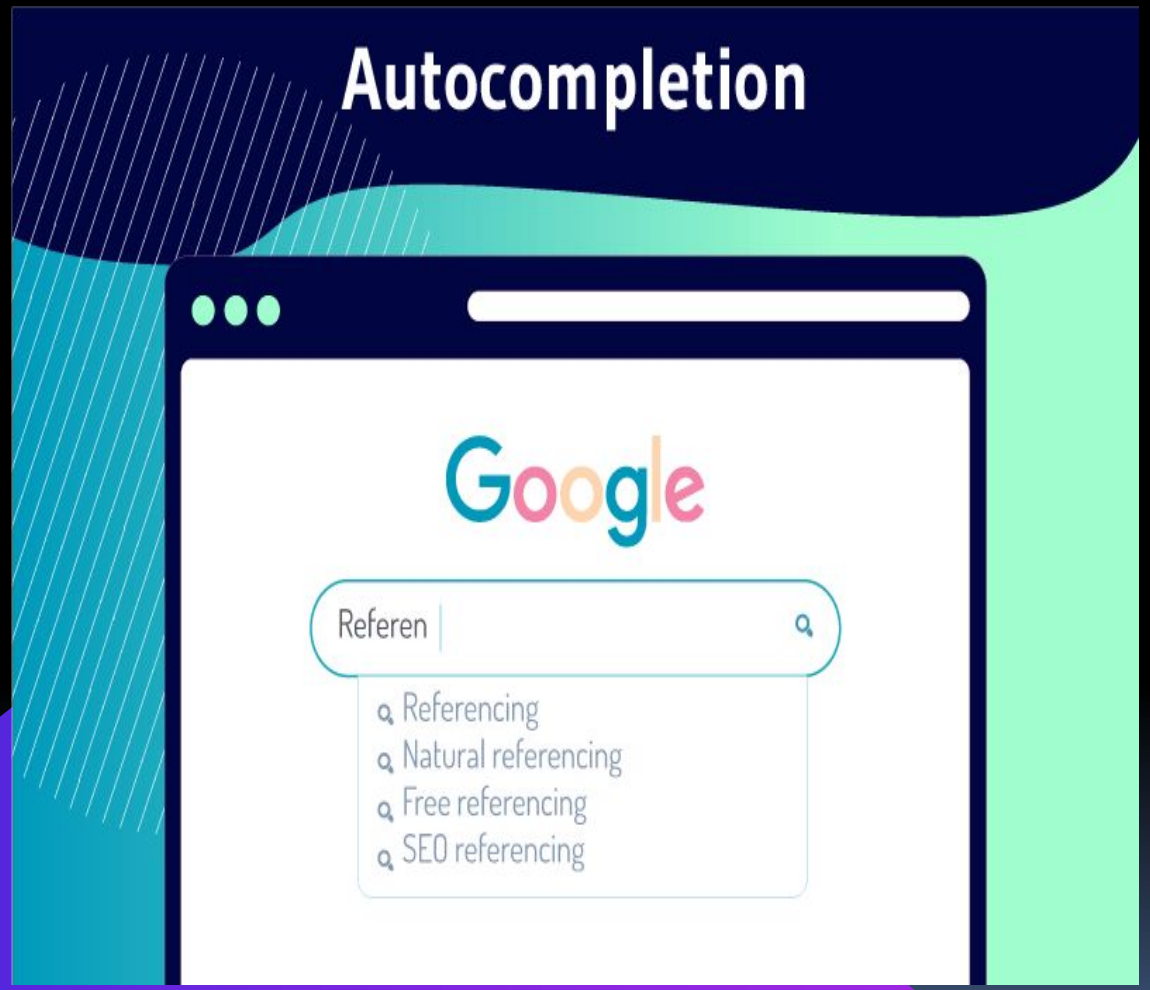


Advantages In Sentence Auto complete

Increased Typing Speed, Enhanced Productivity, Improved Accuracy, Contextual Relevance, User Convenience, Reduced Cognitive Load, Personalization, Language Support, Consistency, Accessibility

Abstract

- The Sentence Autocomplete project aims to create a system that enhances text input interfaces by predicting and suggesting the next words or phrases based on the context of the current input.
- By utilizing advanced machine learning models and NLP techniques, the system can significantly improve typing speed and user experience.
- This document details the algorithm, architecture, implementation, and potential applications of the Sentence Autocomplete system.



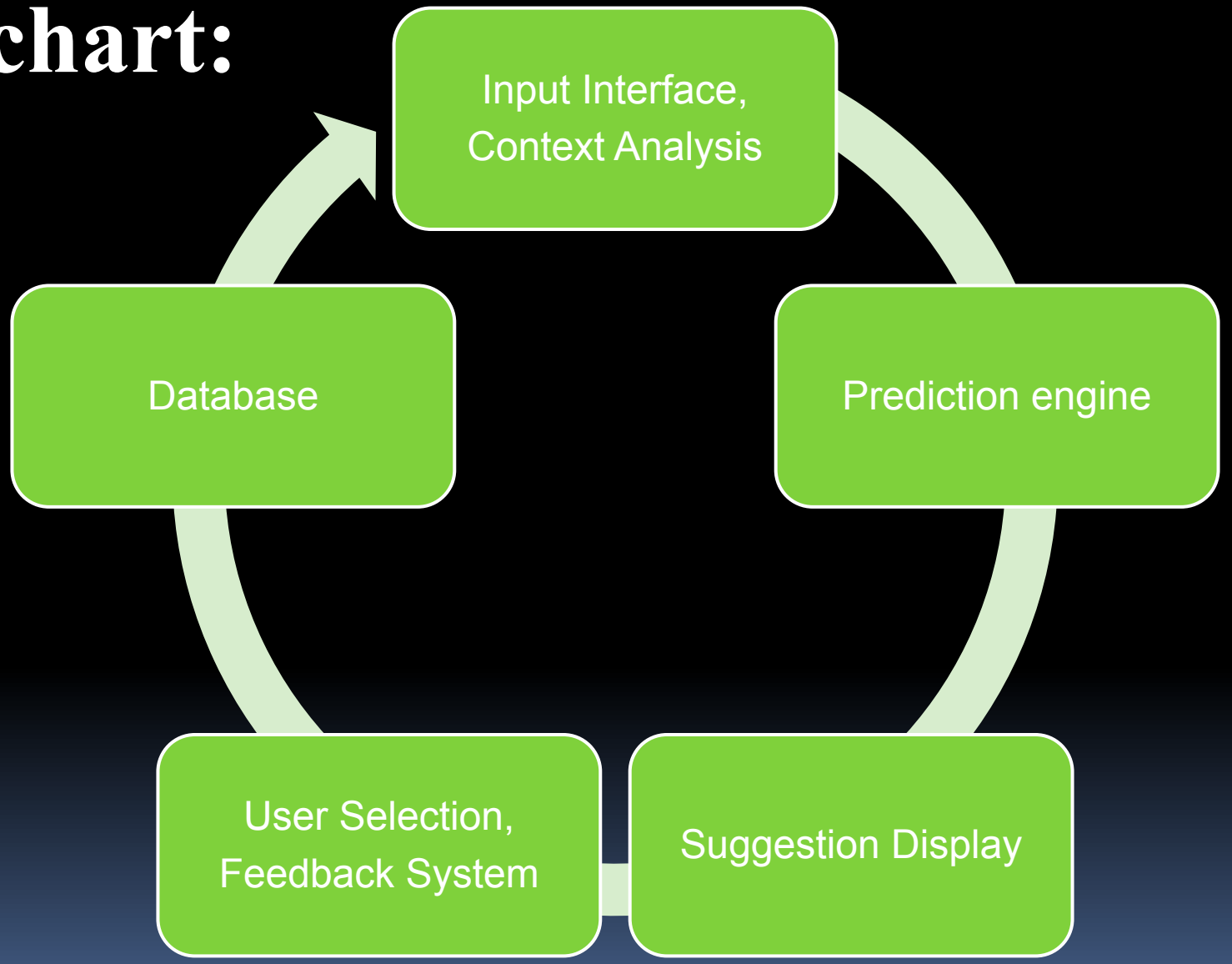
Role Of NLP

- NLP plays a crucial role in sentence autocomplete by understanding the context of the current text to predict relevant next words or phrases.
- NLP adapts to individual user writing styles for personalized suggestions, provides synonyms and alternative phrases, and supports multiple languages.
- By understanding the meaning behind words through semantic analysis, it offers contextually appropriate suggestions and completes entire phrases for more efficient typing.

Real time uses:

- Real-time uses of sentence autocomplete with NLP include email composition
- messaging apps
- coding assistants
- document editing
- search engines
- customer support
- social media
- accessibility tools
- language learning apps
- chatbots.

Flowchart:


Input Interface,
Context Analysis
Prediction engine
Suggestion Display
User Selection,
Feedback System
Database

Challenges And Limitations

- Context Understanding
- Handling Ambiguity
- Personalization
- Language Variations
- Real-Time Processing
- Data Privacy
- Bias in Predictions
- User Trust
- Complex Sentence
- Structures
- Resource Intensive

```
import re
from collections import defaultdict, Counter

class SentenceAutocomplete:
    def __init__(self, corpus):
        self.word_counts = Counter()
        self.word_followers = defaultdict(Counter)
        self.preprocess(corpus)

    def preprocess(self, corpus):
        # Tokenize the corpus into words
        tokens = re.findall(r'\b\w+\b', corpus.lower())

        # Count occurrences of each word and word pairs
        for i in range(len(tokens) - 1):
            self.word_counts[tokens[i]] += 1
            self.word_followers[tokens[i]][tokens[i + 1]] += 1

    def suggest(self, prefix, max_suggestions=5):
        # Get suggestions based on the last word in the prefix
        last_word = prefix.split()[-1].lower()
        suggestions = self.word_followers[last_word].most_common(max_suggestions)
        return [word for word, _ in suggestions]

# Example corpus
corpus = """
The quick brown fox jumps over the lazy dog.
The quick brown fox is very quick and very brown.
Lazy dogs are not quick but they are very lazy.
"""

# Initialize the autocomplete system
autocomplete = SentenceAutocomplete(corpus)

# User input
input_text = input("Start typing your sentence: ")

# Get suggestions
suggestions = autocomplete.suggest(input_text)
print("Suggestions:", suggestions)
```

Execution And output:

Start typing your sentence: The quick
Suggestions: ['brown', 'and', 'is']

Conclusion

- This project demonstrates a basic sentence autocomplete system using NLP, providing word suggestions based on bigram frequencies.
- It's a simple yet effective way to enhance text input efficiency.

References:

- Shaikh, Aryaan, et al. "Autocomplete recommendation plugin and Summarizing Text using Natural Language Processing." *Journal of Innovation Information Technology and Application (JINITA)* 5.2 (2023): 114-123.
- Lee, Mina, Tatsunori B. Hashimoto, and Percy Liang. "Learning autocomplete systems as a communication game." *arXiv preprint arXiv:1911.06964* (2019).
- Shaikh, A., Newalkar, N., Gaikwad, S., Kadav, N. and Shewale, C., 2023. Autocomplete recommendation plugin and Summarizing Text using Natural Language Processing. *Journal of Innovation Information Technology and Application (JINITA)*, *5*(2), pp.114-123.


Thank
You