Natural Language Processing (NLP)

Natural Language Processing (NLP)

- It is a branch of AI and linguistics that helps computers understand, interpret, and use human language—like English, Urdu, Chinees or any spoken or written language.
- Focuses on the interaction between computers and human (natural) languages
- Similar as humans learn to read, listen, write, and speak, NLP teaches computers to do the same with our language.

Simple Examples

- Spell check and auto-correct on your phone? That’s NLP.
- Voice assistants, understanding your commands? NLP.
- Google Translate converting English to Urdu? NLP again.
- Chatbots on websites? NLP is behind them.

What NLP tries to do:

1. Understand what people are saying (or writing).
2. Break it down into parts like words, sentences, and meaning.
3. Respond or act accordingly.

NATURAL LANGUAGE PROCESSING (NLP)

A field of AI focused on enabling computers to understand and process human language

KEY GOALS

- Understand language
- Extract information
- Generate resnonses

COMMON TASKS

- Tokenization
- Part-of-speech tagging
- Named entity recognition
- Sentiment analysis
- Machine translation
- Text classification
- Question answering
- Text summarizzation

Natural Language Processing (NLP)

Goal is to enable machines to read, interpret, generate, and understand human language.

Core Tasks:

- Text classification
- Tokenization and lemmatization
- Named entity recognition (NER)
- Machine translation
- Sentiment analysis
- Text summarization
- Language modeling

NLP Major Working Steps

1. Text Input
2. Text Preprocessing
3. Feature Extraction / Representation
4. NLP Task Execution
5. Output Generation

Text Input

The process starts with raw input, such as:

- A sentence
- A document
- Speech converted to text

Text Preprocessing

Clean and prepare the text for analysis.

Major steps:

- Tokenization: Break text into words or sentences.
  "NLP is fun!" → ["NLP", "is", "fun", "!"]
- Lowercasing: Convert all text to lowercase.
- Removing punctuation/stop words: Eliminate irrelevant tokens like “the”, “is”.
- Stemming/Lemmatization: Reduce words to their base form. "running", "ran" → "run"

Feature Extraction / Representation

Convert text into numbers for machines to understand.

Common methods:

- Bag of Words (BoW)
- TF-IDF (Term Frequency-Inverse Document Frequency)
- Word Embeddings (Word2Vec, GloVe)
- Contextual embeddings (BERT, RoBERTa)

BERT (Bidirectional Encoder Representations from Transformers)
RoBERTa (Robustly Optimized BERT Approach)

NLP Task Execution

The machine performs the required task using models (ML or deep learning):

Examples:

- Sentiment Analysis
- Text Classification
- Machine Translation
- Named Entity Recognition (NER)

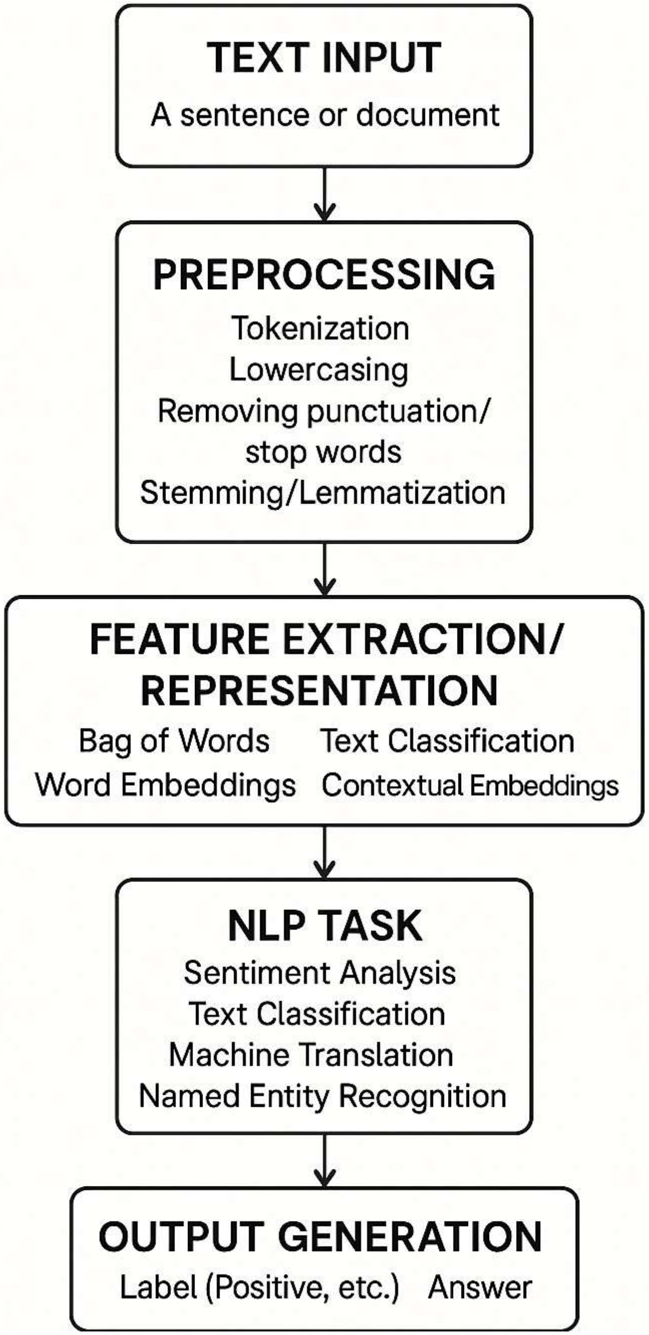
NER: NLP task where a system automatically identifies and classifies named entities in text into predefined categories such as:
Person (e.g., "Elon Musk")
Organization (e.g., "Google")
Location (e.g., "New York")
Date/Time (e.g., "January 1, 2025")
Money (e.g., "$100")

Output Generation

Result is produced and returned to the user:

- Labels (Positive/Negative)
- Translated sentence
- Extracted entities
- Answers to questions

HOW NATURAL LANGUAGE PROCESSING (NLP) WORKS



TEXT INPUT

A sentence or document

PREPROCESSING

Tokenization
Lowercasing
Removing punctuation/ stop words
Stemming/Lemmatization

FEATURE EXTRACTION/ REPRESENTATION

Bag of Words
Text Classification
Word Embeddings
Contextual Embeddings

NLP TASK

Sentiment Analysis
Text Classification
Machine Translation
Named Entity Recognition

OUTPUT GENERATION

Label (Positive, etc.)
Answer

Information Extraction (IE)

IE involves automatically extracting structured information from unstructured or semi-structured text.

It transforms raw text into a machine-readable format.

Main Subtasks:

- Named Entity Recognition (NER): Identifying proper nouns like people, organizations, places.
- Relation Extraction: Identifying semantic relationships (e.g., “Alice works at OpenAI” → works_for(Alice, OpenAI)).
- Event Extraction: Detecting events and their participants in text.

Example:

“John Doe joined Google in 2015.”

Person: John Doe

Organization: Google

Date: 2015

Relation: joined

Question Answering (QA)

- QA systems are designed to automatically answer questions posed in natural language. They can be:
  - Closed-domain: Specialized in a specific field (e.g., medical QA)
  - Open-domain: Answers general questions using large corpora (e.g., search engines, LLMs)

Approaches:

Retrieval-Based QA: Find answers from documents (e.g., Wikipedia + BERT).

Generative QA: Generate answers using language models (e.g., GPT, T5).

Example:

Question: “Who discovered penicillin?”

Answer: “Alexander Fleming.”

Generative Pre-trained Transformer- GPT is a language model created by OpenAI. It reads a lot of text (pretraining), learns patterns, and can generate human-like text

T5 is a model by Google that converts all NLP tasks into a text-to-text format

NLP in Information Retrieval (IR)

IR involves finding relevant documents or data from a large collection.

NLP enhances IR by enabling better understanding of both queries and documents.

Key Applications:

- Search engines (e.g., Google, Bing)
- Question answering systems
- Document ranking using transformers (e.g., ColBERT, Dense Passage Retrieval)

ColBERT (Contextualized Late Interaction over BERT), *designed for tasks like document retrieval, question answering, and semantic search*

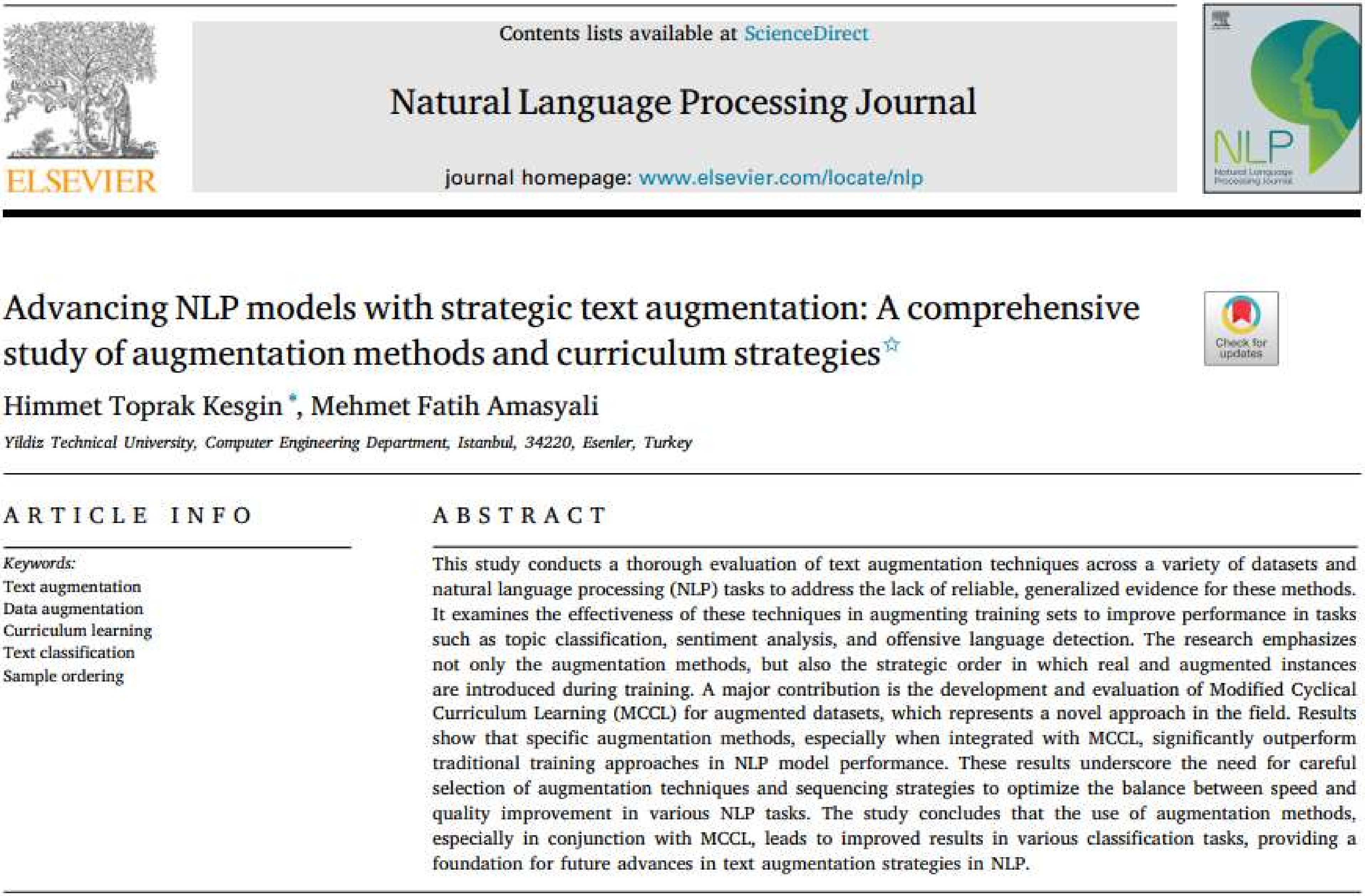
NLP Techniques in IR:

- Query expansion
- Relevance feedback
- Semantic similarity via embeddings (Word2Vec, BERT)

Morphology in NLP

- Morphology is the study of word formation and structure in a language.
- It helps NLP systems understand how words are constructed and inflected.
- **Types:**
  - Inflectional Morphology: Alters a word’s form to express grammatical features (e.g., run → runs).
  - Derivational Morphology: Creates new words (e.g., happy → happiness).
- **Applications in NLP:**
  - Lemmatization: Mapping words to their base forms.
  - Morphological analysis: Essential for morphologically rich languages (e.g., Arabic, Finnish).
  - Tokenization and subword modeling (e.g., Byte Pair Encoding in BERT).

https://doi.org/10.48550/arXiv.2406.12618



Contents lists available at ScienceDirect

Natural Language Processing Journal

journal homepage: www.elsevier.com/locate/nlp

Advancing NLP models with strategic text augmentation: A comprehensive study of augmentation methods and curriculum strategies☆

Himmet Toprak Kesgin *, Mehmet Fatih Amasyali

Yildiz Technical University, Computer Engineering Department, Istanbul, 34220, Esenler, Turkey

ARTICLE INFO

Keywords:
Text augmentation
Data augmentation
Curriculum learning
Text classification
Sample ordering

ABSTRACT

This study conducts a thorough evaluation of text augmentation techniques across a variety of datasets and natural language processing (NLP) tasks to address the lack of reliable, generalized evidence for these methods. It examines the effectiveness of these techniques in augmenting training sets to improve performance in tasks such as topic classification, sentiment analysis, and offensive language detection. The research emphasizes not only the augmentation methods, but also the strategic order in which real and augmented instances are introduced during training. A major contribution is the development and evaluation of Modified Cyclical Curriculum Learning (MCCL) for augmented datasets, which represents a novel approach in the field. Results show that specific augmentation methods, especially when integrated with MCCL, significantly outperform traditional training approaches in NLP model performance. These results underscore the need for careful selection of augmentation techniques and sequencing strategies to optimize the balance between speed and quality improvement in various NLP tasks. The study concludes that the use of augmentation methods, especially in conjunction with MCCL, leads to improved results in various classification tasks, providing a foundation for future advances in text augmentation strategies in NLP.

Natural Language Processing (NLP)

Interpretability and Analysis (IA) research in Natural Language Processing (NLP) aims to deepen our understanding of the behavior and inner workings of NLP systems.

Despite its growth, IA research faces criticism for lacking actionable insights that directly influence the development of new NLP models.

This paper seeks to quantify the impact of IA research on the broader NLP field through a mixed-methods approach, combining bibliometric analysis and community surveys.

Interpretability refers to how easily a human can understand the internal workings or predictions of a machine learning model — especially why a model made a particular decision.

Interpretability

Interpretability refers to how easily a human can understand the internal workings or predictions of a machine learning model — especially why a model made a particular decision.

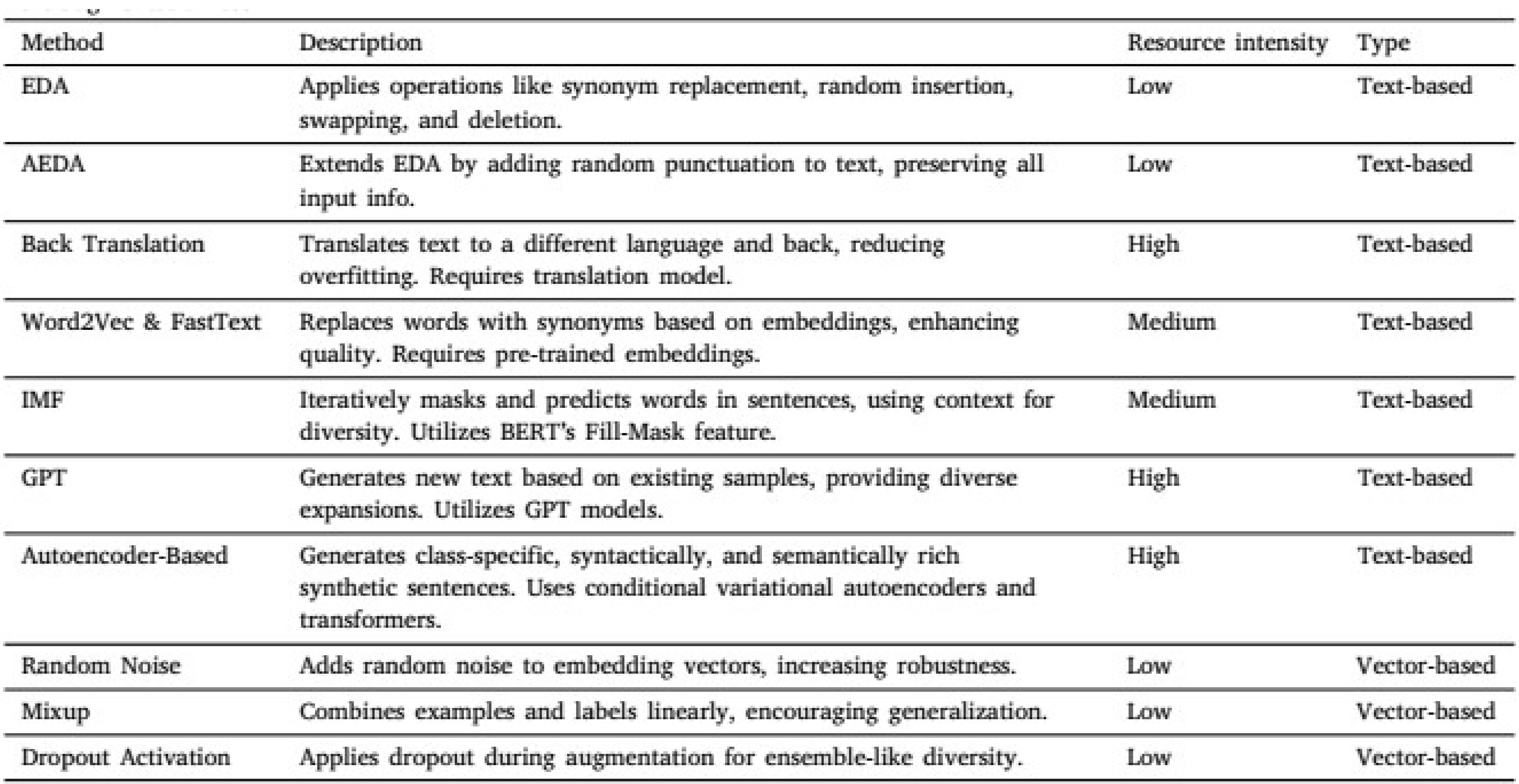
Trust: Users and stakeholders can trust models they understand.

Debugging: Helps identify model errors, biases, or flaws.

Compliance: Essential in regulated industries (e.g., finance, healthcare).

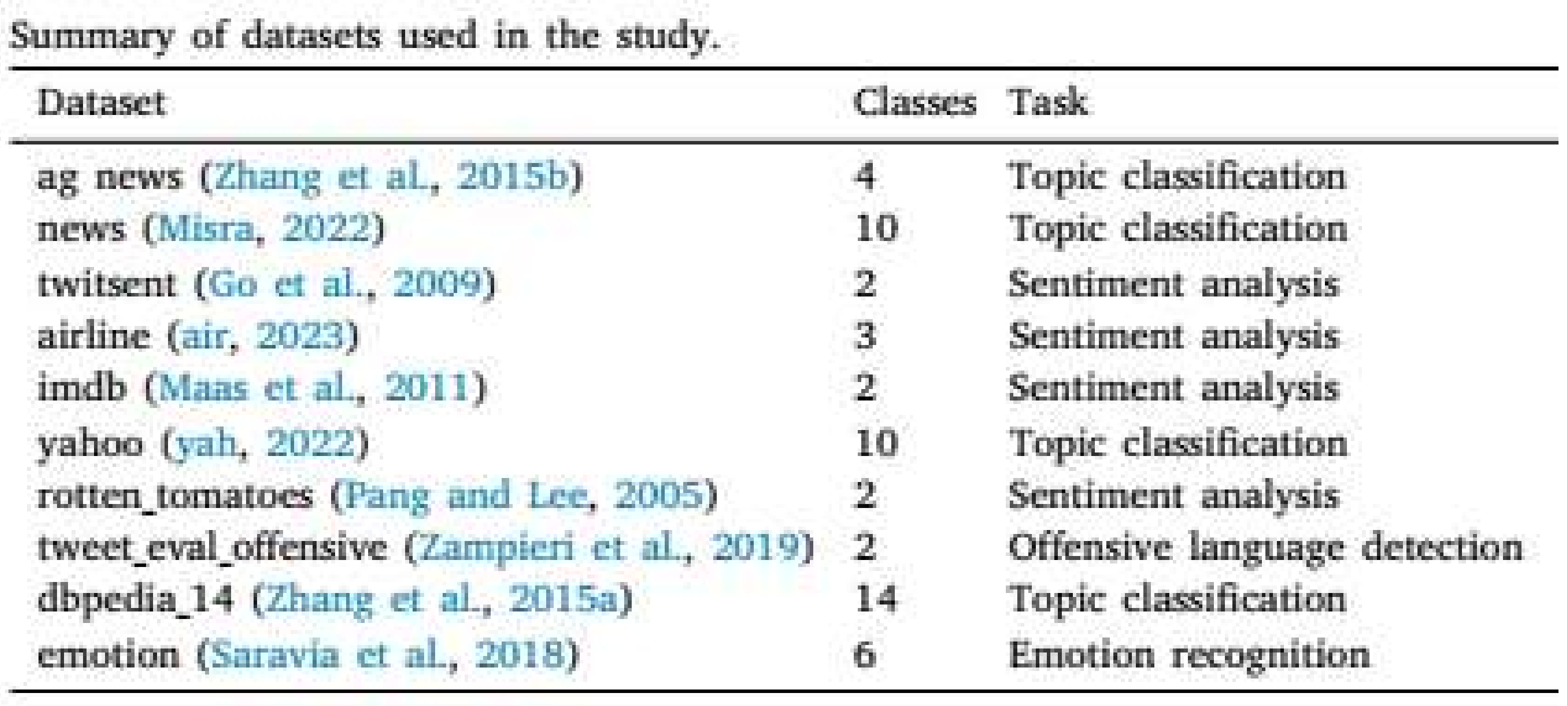
Fairness: Detects and mitigates bias or unethical behavior.

Text augmentation techniques



Method	Description	Resource intensity	Type
EDA	Applies operations like synonym replacement, random insertion, swapping, and deletion.	Low	Text-based
AEDA	Extends EDA by adding random punctuation to text, preserving all input info.	Low	Text-based
Back Translation	Translates text to a different language and back, reducing overfitting. Requires translation model.	High	Text-based
Word2Vec & FastText	Replaces words with synonyms based on embeddings, enhancing quality. Requires pre-trained embeddings.	Medium	Text-based
IMF	Iteratively masks and predicts words in sentences, using context for diversity. Utilizes BERT's Fill-Mask feature.	Medium	Text-based
GPT	Generates new text based on existing samples, providing diverse expansions. Utilizes GPT models.	High	Text-based
Autoencoder-Based	Generates class-specific, syntactically, and semantically rich synthetic sentences. Uses conditional variational autoencoders and transformers.	High	Text-based
Random Noise	Adds random noise to embedding vectors, increasing robustness.	Low	Vector-based
Mixup	Combines examples and labels linearly, encouraging generalization.	Low	Vector-based
Dropout Activation	Applies dropout during augmentation for ensemble-like diversity.	Low	Vector-based

Datasets used in the Study



Summary of datasets used in the study.

Dataset	Classes	Task
ag news (Zhang et al., 2015b)	4	Topic classification
news (Misra, 2022)	10	Topic classification
twitsent (Go et al., 2009)	2	Sentiment analysis
airline (air, 2023)	3	Sentiment analysis
imdb (Maas et al., 2011)	2	Sentiment analysis
yahoo (yah, 2022)	10	Topic classification
rotten_tomatoes (Pang and Lee, 2005)	2	Sentiment analysis
tweet_eval_offensive (Zampieri et al., 2019)	2	Offensive language detection
dbpedia_14 (Zhang et al., 2015a)	14	Topic classification
emotion (Saravia et al., 2018)	6	Emotion recognition

Methodology

The study employs a mixed-methods analysis comprising:

1. Bibliometric Analysis:

- Constructed a citation graph of over 185,000 papers from ACL and EMNLP conferences (2018–2023).
- Assessed the centrality and citation patterns of IA papers within the NLP literature.

2. Community Survey:

- Surveyed 138 NLP researchers and practitioners to gather qualitative insights into the perceived impact and utility of IA research.

3. Manual Annotation:

- Conducted a qualitative analysis of 556 papers to understand how IA findings influence subsequent research.

Finding

1. Growth of IA Research:

- IA papers increased from 90 in 2020 to 160 in 2023, marking a 77.8% growth rate, the highest among NLP subfields.

2. Citation Impact:

- IA work is well-cited outside its subfield and holds a central position in the NLP citation graph.
- Many novel methods are proposed based on IA findings, indicating their influence on the development of new approaches.

3. Perceptions from the NLP Community:

- NLP researchers build upon IA findings and perceive them as crucial for progress across multiple subfields.
- IA research is valued for providing terminology and frameworks that aid in understanding and developing NLP models.

4. Limitations Identified:

- While IA findings are cited, highly influential non-IA work often references IA research without being directly driven by it.
- There is a need for IA research to offer more actionable insights that can directly inform model development.

Future IA Research Directions

To enhance the impact of IA research, the authors suggest:

1. Unification:

- Develop standardized frameworks and methodologies to consolidate IA research efforts.

2. Actionable Recommendations:

- Focus on producing insights that can directly inform the design and improvement of NLP models.

3. Human-Centered, Interdisciplinary Work:

- Collaborate with experts from other disciplines to ensure IA research addresses real-world needs and applications.

4. Standardized, Robust Methods:

- Employ rigorous evaluation metrics and methodologies to strengthen the reliability of IA findings.

Conclusion

Interpretability and Analysis research plays a central role in the NLP field, both in terms of scholarly influence and community perception.

However, to maximize its impact, IA research must evolve to provide more actionable, standardized, and interdisciplinary insights that directly contribute to the advancement of NLP technologies.

Summary of the Paper

This study presents an in-depth evaluation of various text augmentation techniques across multiple NLP tasks, including topic classification, sentiment analysis, and offensive language detection.

Recognizing the challenges posed by data sparsity and the scarcity of labeled data in NLP, the authors aim to provide generalized evidence on the effectiveness of augmentation methods.

A significant contribution of the paper is the introduction of Modified Cyclical Curriculum Learning (MCCL), a novel training strategy that emphasizes the strategic sequencing of real and augmented data during model training.

The authors demonstrate that integrating specific augmentation methods with MCCL can significantly enhance model performance compared to traditional training approaches.

Methodology

1. Dataset Selection and Preprocessing

- Ten diverse datasets covering various NLP tasks were selected.
- Each dataset was partitioned into training and test sets using stratified sampling.
- Preprocessing steps included converting text to lowercase and truncating sentences to the first 300 characters.
- For the TwitSent dataset, a smaller training set of 500 samples was used due to initial observations on statistical significance.

2. Text Representation and Modeling

The BERT model was employed for text representation, transforming each text sample into a 768-dimensional vector.

3. Text Augmentation Techniques

The study evaluated a range of text augmentation methods:

- Easy Data Augmentation (EDA): Involves synonym replacement, random insertion, random swapping, and random deletion.
- An Easier Data Augmentation (AEDA): Adds random punctuation to text.
- Back Translation: Translates text to another language and back to the original.
- Word2Vec/FastText Replacement: Replaces words with synonyms based on word embeddings.
- Iterative Mask Filling (IMF): Iteratively masks and predicts words in sentences using BERT or TinyBERT.

4. Curriculum Learning Strategy

- The authors introduced Modified Cyclical Curriculum Learning (MCCL), which involves the strategic sequencing of real and augmented data during training.
- MCCL cycles through different subsets of data, gradually increasing complexity to enhance model learning.

5. Evaluation Metrics

- Model performance was assessed using standard metrics such as accuracy and F1-score.
- Comparisons were made between models trained with and without augmentation, as well as with and without MCCL.

Question on paper

You are leading the NLP team at a startup developing a sentiment analysis system for customer feedback in a specialized e-commerce domain (e.g., handmade luxury goods).

The dataset consists of only 2,000 labeled reviews and 10,000 unlabeled ones.

The domain includes nuanced vocabulary (e.g., artisan terms, cultural expressions), and the initial BERT-based model shows low F1 scores on minority sentiment classes (e.g., “mixed” or “disappointed”).

You are tasked with improving model performance using text augmentation and curriculum learning, based on insights from the 2024 paper by Kesgin and Amasyali.

Constraints

Limited Compute: You must train and fine-tune your model on a single mid-range GPU (e.g., NVIDIA RTX 3060) within 8 hours total training time.

Augmentation Budget: You are allowed to generate no more than 2× the original labeled data through augmentation.

Model Consistency: The augmented data must not significantly alter the original sentiment label; semantic preservation is crucial.

No Human-in-the-Loop: Due to cost constraints, your team cannot manually validate augmented samples.

Task

As the team lead, design an augmentation-driven pipeline under these constraints. Address the following:

Augmentation Strategy:

- Which methods (e.g., synonym replacement, back-translation, EDA, contextual augmentation) would you choose and why?
- How will you ensure semantic consistency without human validation?

Curriculum Design:

- Propose a sequencing approach for feeding the data (original and augmented) into the model.
- How will curriculum learning help mitigate overfitting on high-frequency sentiment patterns?

Risk Mitigation:

- Identify two risks of using synthetic data in this setting and explain how you would control for them.

Evaluation Plan:

- How will you measure improvement beyond accuracy (e.g., macro F1, confusion matrix)?
- Suggest a way to analyze whether curriculum learning made a meaningful difference.

Solution

Augmentation Strategy

I would use a combination of:

- Synonym Replacement (via WordNet or domain-specific embedding similarity),
- Contextual Augmentation (e.g., BERT masked token prediction).

Justification:

- These are light-weight and align with the 2× augmentation limit and compute constraint.
- They tend to preserve semantics better than paraphrasing or back-translation.

To maintain semantic consistency:

- I would pass all augmented samples through a pretrained sentiment classifier. If the predicted label differs from the original, the sample is discarded.
- I'd also restrict replacements to terms occurring in our domain corpus to avoid jargon drift.

Curriculum Design

I would:

- Begin with original, high-confidence samples (e.g., top 50% by classifier agreement).
- Gradually mix in augmented samples, ordered by predicted confidence or lexical simplicity (shorter, simpler first).
- Delay introduction of low-frequency class samples to stabilize early training and reduce overfitting.

Why it works:

- Helps the model stabilize before handling noise and rare class data.
- Encourages progressive learning, aligned with human-like educational principles.

Risk Mitigation

1. Risk: Semantic drift in augmentation.

    Mitigation: Filter using a sentiment classifier to verify label consistency.

2. Risk: Out-of-domain vocabulary pollution.

    Mitigation: Limit augmentation choices to known vocabulary or embeddings close to in-domain words.

Evaluation Plan

- Use macro F1-score to ensure fairness across all sentiment classes.
- Create a confusion matrix to assess where misclassifications occur (e.g., neutral ↔ mixed).
- Conduct an ablation study:
    - ➢ Compare model trained with curriculum vs. without.
    - ➢ Evaluate convergence speed, F1-scores, and per-class performance.