Novelty Assessment Report

Paper: Tokenisation over Bounded Alphabets is Hard
PDF URL: https://openreview.net/pdf?id=Xhf9YqwlM4
Venue: ICLR 2026 Conference Submission
Year: 2026
Report Generated: 2026-01-05

Abstract

Recent works have proven tokenisation to be NP-complete. However, their proofs' constructions rely on tokenisation being applied to inputs with alphabets of unbounded cardinality, which does not accurately reflect the real world. Indeed, since practical applications of tokenisers involve fixed-size alphabets (e.g., Unicode or bytes), the implications of such a statement may be challenged. In this work, we examine the computational complexity of tokenisation over bounded alphabets, considering two variants of this problem: bottom-up tokenisation and direct tokenisation, where we must, respectively, select a sequence of merge operations (in bottom-up tokenisation) or a vocabulary (in direct tokenisation) whose application compresses a dataset to at most δ symbols. When alphabets are bounded to have only 2 characters, we do not only prove that bottom-up and direct tokenisation are NP-complete, but also that there is no polynomial-time approximation scheme for either of these problems (unless P = NP). Furthermore, even when alphabets are bounded to contain a single character, we can still prove the NP-completeness of direct tokenisation. Although the single-character case is not practical on its own, proving hardness results for an N-ary alphabet allows us to prove the same results for alphabets of any larger size. We thus conclude that direct tokenisation over any alphabet is NP-complete, and that both bottom-up and direct tokenisation do not admit polynomial-time approximation schemes for any alphabet of size 2 or larger.

Core Task Landscape

This paper addresses: **Computational Complexity of Tokenisation over Bounded Alphabets**

A total of **7 papers** were analyzed and organized into a taxonomy with **7 categories**.

Taxonomy Overview

The research landscape has been organized into the following main categories:

- **Theoretical Complexity Analysis of Tokenisation**
- **Algorithm Design and Optimization**
- **Application Domains and Empirical Studies**

Complete Taxonomy Tree

- Computational Complexity of Tokenisation over Bounded Alphabets Survey Taxonomy
- Theoretical Complexity Analysis of Tokenisation
  - NP-Completeness Proofs for Bounded Alphabets ★ (1 papers)
  - [0] Tokenisation over Bounded Alphabets is Hard (Anon et al., 2026) View paper
- Algorithm Design and Optimization
  - Byte Pair Encoding Variants (2 papers)
  - [1] MorphTok: Morphologically Grounded Tokenization for Indian Languages (Maharaj Brahma, 2025) View paper
  - [3] Byte Pair Encoding Dictionary Equivalence (Bergmark, 2025) View paper
  - Security-Preserving Tokenisation (1 papers)
  - [2] FAST: secure and high performance format-preserving encryption and tokenization (F. B. Durak, 2021) View paper
  - Alternative Tokenisation Approaches (1 papers)
  - [5] Hidden Markov Model-based approach for Efficient f Lexical Analysis (W Li, n.d.) View paper
- Application Domains and Empirical Studies
  - Network Security and Anomaly Detection (1 papers)
  - [4] Micro-signatures: The Effectiveness of Known Bad N-Grams for Network Anomaly Detection (Richard E. Harang, 2016) View paper
  - Biological Sequence Analysis (1 papers)
  - [6] gRNAdeX: eXpressive, Biologically-eXtensible gRNAde (ML Sancho, n.d.) View paper
  - Language Model Learning and Prediction (1 papers)
  - [7] Pre-trained Large Language Models Learn to Predict Hidden Markov Models In-context (Y Dai, n.d.) View paper

Narrative

Core task: computational complexity of tokenisation over bounded alphabets. This field examines the theoretical and practical challenges of segmenting sequences into meaningful units when the underlying symbol set is finite and constrained. The taxonomy reveals three main branches that together capture the landscape. Theoretical Complexity Analysis of Tokenisation investigates fundamental hardness results, including NP-completeness proofs and decision problems that arise when alphabet size is limited. Algorithm Design and Optimization focuses on developing efficient procedures—ranging from greedy heuristics to dynamic programming solutions—that balance computational cost with segmentation quality. Application Domains and Empirical Studies explore how tokenisation methods perform in real-world settings such as natural language processing, bioinformatics, and cryptographic systems, often drawing on works like MorphTok[1] for morphological segmentation or FAST Encryption[2] for secure encoding schemes.

Several active lines of work highlight contrasting priorities and open questions. One thread examines equivalence and optimality criteria, as seen in BPE Dictionary Equivalence[3], which studies when different tokenisation schemes yield identical outcomes. Another thread addresses pattern recognition and probabilistic models, exemplified by HMM Lexical Analysis[5] and LLMs Learn HMMs[7], which probe how statistical frameworks can guide segmentation under bounded alphabets. The original paper, Tokenisation Bounded Alphabets[0], sits squarely within the theoretical complexity branch, specifically targeting NP-completeness proofs for bounded alphabets. Its emphasis on rigorous hardness results distinguishes it from more algorithm-centric studies like BPE Dictionary Equivalence[3], which prioritize structural properties over worst-case complexity, and from application-oriented works such as gRNAdeX[6] or Micro-signatures[4], which focus on domain-specific performance rather than foundational computational limits.

Related Works in Same Category

No sibling papers and no sibling subtopics were found under the same parent taxonomy node; the paper appears structurally isolated in the taxonomy.

Contributions Analysis

Overall novelty summary. The paper establishes NP-completeness and inapproximability results for tokenisation over bounded alphabets, specifically proving hardness for binary (2-character) and unary (1-character) cases. Within the taxonomy, it occupies the sole position in the 'NP-Completeness Proofs for Bounded Alphabets' leaf under 'Theoretical Complexity Analysis of Tokenisation'. This leaf contains only the original paper itself, indicating a sparse research direction. The broader parent branch ('Theoretical Complexity Analysis') contains just two leaves, with the sibling 'Inapproximability Results' leaf currently empty, suggesting this theoretical angle remains relatively unexplored in the literature.

The taxonomy reveals that most tokenisation research concentrates in 'Algorithm Design and Optimization' (3 leaves: BPE variants, security-preserving methods, alternative approaches) and 'Application Domains' (3 leaves: network security, biological sequences, language models). The original paper's theoretical focus contrasts sharply with these algorithm-centric and application-oriented directions. Neighboring work like BPE Dictionary Equivalence examines structural properties rather than worst-case complexity, while HMM Lexical Analysis and LLMs Learn HMMs explore probabilistic segmentation models. The taxonomy's scope notes explicitly separate theoretical hardness proofs from practical algorithm design, positioning this work in a distinct, less-populated research space.

Among the three contributions analyzed, the literature search examined only 5 candidate papers total. For the binary tokenisation NP-completeness result, 1 candidate was examined with 0 refutations. The unary direct tokenisation hardness examined 1 candidate with 0 refutations. The formal problem definitions examined 3 candidates with 0 refutations. This limited search scope—covering top-K semantic matches plus citation expansion—found no prior work that clearly overlaps with the specific bounded-alphabet hardness results. All three contributions appear novel within the examined candidate set, though the small search scale (5 papers) means the analysis cannot claim exhaustive coverage of the theoretical complexity literature.

Given the sparse taxonomy position and absence of refuting candidates among the 5 papers examined, the work appears to address a relatively unexplored theoretical question. However, the limited search scope and the fact that the original paper is the only entry in its taxonomy leaf suggest caution: the analysis reflects top-K semantic similarity rather than comprehensive field coverage. The novelty assessment is constrained by what was examined, not by what exists in the broader literature.

This paper presents **3 main contributions**, each analyzed against relevant prior work:

Contribution 1: NP-completeness of binary tokenisation and hardness of approximation

Description: The authors prove that both bottom-up and direct tokenisation over binary alphabets are NP-complete decision problems and establish that neither problem admits a polynomial-time approximation scheme, showing they cannot be approximated arbitrarily well in polynomial time.

This contribution was assessed against **1 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. Restricted Common Superstring and Restricted Common Supersequence

URL: View paper

Brief Assessment

Restricted Common Superstring[8] addresses string permutation problems (finding optimal orderings of multisets to maximize substring/subsequence matches), not tokenisation problems involving vocabulary selection or merge operations for text compression.

Contribution 2: Strong NP-completeness of direct unary tokenisation

Description: The authors establish that direct tokenisation remains NP-complete even for the simplest case of unary alphabets (single character). They prove strong NP-hardness by reduction from the vertex cover problem, showing hardness holds even when strings are explicitly represented.

This contribution was assessed against **1 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. The complexity of some regex crossword problems

URL: View paper

Brief Assessment

Regex Crossword Complexity[9] addresses NP-completeness of regex crossword puzzles over binary alphabets, not tokenisation problems. The technical domains are entirely different—one concerns pattern matching in grid puzzles, the other concerns string compression and vocabulary selection.

Contribution 3: Formal definition of n-ary tokenisation problems over bounded alphabets

Description: The authors introduce and formalize the n-ary tokenisation problem, defining decision and optimisation variants for both direct and bottom-up tokenisation over alphabets constrained to size n, establishing a framework for analyzing tokenisation complexity with bounded alphabets.

This contribution was assessed against **3 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. Description and analysis of a bottom-up DFA minimization algorithm

URL: View paper

Brief Assessment

Bottom-up DFA Minimization[10] addresses deterministic finite automaton minimization using strongly connected components, not tokenisation complexity over bounded alphabets. The paper focuses on graph-theoretic algorithms for automata rather than natural language processing tokenisation problems.

2. FAST: secure and high performance format-preserving encryption and tokenization

URL: View paper

Brief Assessment

FAST Encryption[2] addresses format-preserving encryption and tokenization for data protection, not the computational complexity analysis of tokenisation algorithms over bounded alphabets that is the focus of the original paper.

3. A BOTTOM-UP MINIMIZATION DFA ALGORITHM AND APPLICATIONS

URL: View paper

Brief Assessment

Bottom-up DFA Algorithm[11] focuses on minimizing deterministic finite automata through bottom-up processing of strongly connected components, not on tokenisation complexity over bounded alphabets. The paper addresses automata minimization algorithms, which is a fundamentally different computational problem from tokenisation.

Appendix: Text Similarity Detection

No high-similarity text segments were detected across any compared papers.

References

- [0] Tokenisation over Bounded Alphabets is Hard View paper
- [1] MorphTok: Morphologically Grounded Tokenization for Indian Languages View paper
- [2] FAST: secure and high performance format-preserving encryption and tokenization View paper
- [3] Byte Pair Encoding Dictionary Equivalence View paper
- [4] Micro-signatures: The Effectiveness of Known Bad N-Grams for Network Anomaly Detection View paper
- [5] Hidden Markov Model-based approach for Efficient f Lexical Analysis View paper
- [6] gRNAdeX: eXpressive, Biologically-eXtensible gRNAde View paper
- [7] Pre-trained Large Language Models Learn to Predict Hidden Markov Models In-context View paper
- [8] Restricted Common Superstring and Restricted Common Supersequence View paper
- [9] The complexity of some regex crossword problems View paper
- [10] Description and analysis of a bottom-up DFA minimization algorithm View paper
- [11] A BOTTOM-UP MINIMIZATION DFA ALGORITHM AND APPLICATIONS View paper