INTRODUCTION

- Tajik (Cyrillic) is severely under-resourced for NLP.
- Existing Persian tools (BidNLP, DadmaTools) cannot process Cyrillic.
- Transliteration research does not provide full linguistic analysis.
- **Goal:** First comprehensive open-source pipeline for Tajik Cyrillic text, combining classical and neural methods.

RELATED WORK

- **Persian libraries:** BidNLP, Shekar, DadmaTools, DadmaTools V2 Arabic script only, cannot process Cyrillic.
- **Transliteration:** statistical (Davis, Grashchenko) and neural (Sadraei Javaheri, Merchant et al.) important auxiliary task, but no full pipeline.
- **Low-resource guidelines:** LowResource 2026 TajikNLP follows these principles.
- **Gap:** No holistic toolkit for Tajik Cyrillic until now.

LINGUISTIC RESOURCES (Apache 2.0)

Dataset	Size
POS-tagged corpus (28 tags)	52,508 entries
Sentiment lexicon (polarity + intensity)	3,517 entries
Toponym gazetteer (type, region)	5,559 entries
Personal names (gender, translit.)	3,842 entries

Table: Public datasets on Hugging Face Hub

Sources: textbooks, dictionaries, Tajik National Corpus, official name list, geographic gazetteers.

ARCHITECTURE OVERVIEW

- **Core:** `Doc` object (list of `Tokens` with attributes: lemma, POS, span, metadata).
- **Components** inherit from `BaseComponent`, declare `requires`/`assigns`.
- `TajikPipeline` manages sequential execution and dependency checks.
- Pre-configured pipelines: `"default"`, `"neural"`, `"sentiment"`.

Full component list:

1. `TextCleaner` remove URLs, emails, mentions, normalize whitespace.
2. `TajikCyrillicNormalizer` map non-standard Cyrillic, correct OCR errors, unify punctuation.
3. `RegexTokenizer` extract words, numbers, punctuation, emails, URLs.
4. `MorphemeTokenizer` recursive morpheme splitting (prefix/suffix rules).
5. `TajikSubwordTokenizer` pre-trained BPE for subword segmentation.
6. `RegexSentencizer` sentence boundary detection with abbreviation list.
7. `RuleBasedPOSTagger` dictionary + heuristic suffix stripping (28 POS tags).
8. `DictLemmatizer` / `DictStemmer` hybrid (dictionary + affix rules).
9. **Unified Morphology Engine** three modes: `lemma`, `controlled`, `deep`.
10. `LexiconSentimentAnalyzer` handles negations and intensifiers.
11. `TajikEmbeddings` pre-trained Word2Vec/FastText from Hugging Face.
12. `TajikCountVectorizer` / `TajikTfidfVectorizer` BoW and TF-IDF.
13. `KeywordClassifier` rule-based text classification.
14. `StopWordsFilter` list of 150+ stop words.

UNIFIED MORPHOLOGY ENGINE

- Consolidates prefix and suffix rules into single configurable engine.
- **Three modes:**
  1. `lemma` conservative lemmatisation.
  2. `controlled` broader stripping with POS and dictionary checks.
  3. `deep` aggressive stemming.
- Combines explicit dictionary lookup (≈5,000 high-frequency lemmas) with rule-based affix stripping.
- **7 prefix groups** and **14 suffix groups** (ordered longest to shortest).
- **136 exception entries** suppletive verb stems, irregular plurals, pronoun enclitics.

ARCHITECTURE DIAGRAM

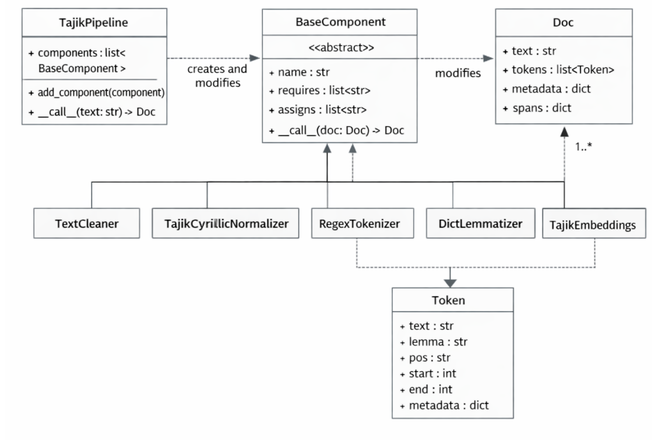

Figure: Main architectural elements and data flow

PIPELINE FLOWCHART

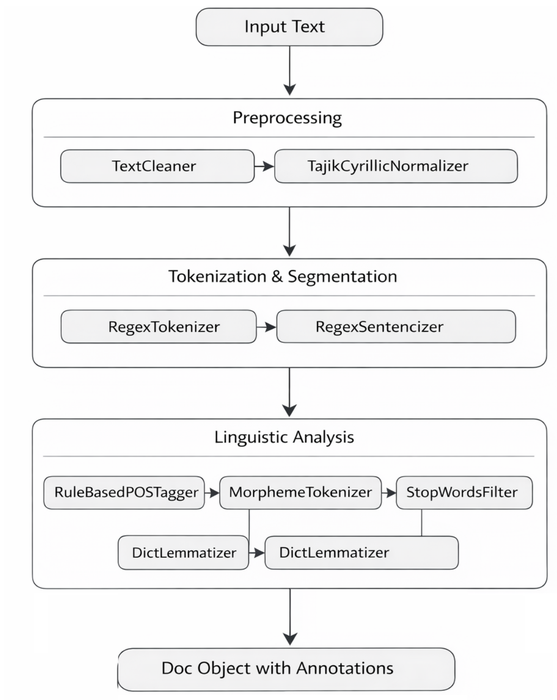

Figure: `"default"` pipeline processing sequence

PERFORMANCE EVALUATION

Component	P	R	F1
Rule-based POS tagger	0.87	0.86	0.86
Unified lemmatiser (controlled)	0.89	0.88	0.88
Stemmer (safe mode)	0.91	0.90	0.90

Table: Tested on 1,500 held-out sentences (≈12k tokens)

Unified lemmatiser outperforms classic hybrid (F1 0.88 vs. 0.82).

Practical impact: Before TajikNLP, processing Tajik Cyrillic required 35 person-months; now it takes minutes.

SENTIMENT ANALYSIS

- Lexicon-based (`tajik-sentiment-lexicon`) with polarity and intensity.
- **Negation handling:** flips polarity, multiplies by 0.8 damping factor (default window: 3 tokens left).
- **Intensifiers:** multiply absolute intensity by factor (1.21.8).
- Combined with Unified Morphology Engine recognises sentiment in inflected forms.
- Examples: ! → +1.05; . → −1.40.

FEATURE EXTRACTION & METRICS

- `TajikCountVectorizer` Bag-of-Words with n-gram support.
- `TajikTfidfVectorizer` TF-IDF matrices.
- `metrics` module: precision, recall, F1-score, Levenshtein distance, WER, BLEU.
- `KeywordClassifier` rule-based text classification.
- `StopWordsFilter` removes functional vocabulary (150+ stop words).

PROCESSING EXAMPLES

Example 1: Lemmatisation & POS tagging

Token	Lemma	POS
		NOUN
		VERB
		VERB

Example 2: Stemming (safe vs. deep)

Word	Stem (safe)	Stem (deep)

Example 3: Neural pipeline & embeddings

BPE segments rare names; `most_similar()` → , , .

Example 4: Keyword-Based Classification

The `KeywordClassifier` assigns texts to user-defined categories. With categories “sports” (, ,) and “politics” (, ,), the sentences . and . are correctly classified as “sports” and “politics”, respectively.

DISCUSSION & COMPARISON

- Persian NLP toolkits (BidNLP, DadmaTools) are restricted to Arabic script.
- Transliteration research does not provide a complete pipeline for native Cyrillic analysis.
- TajikNLP fills this gap: first comprehensive, open-source toolkit for Tajik Cyrillic.
- Modular architecture + unified morphology engine + pre-trained embeddings + openly released datasets.
- High test coverage (93%) and CI ensure production-ready reliability.

LIMITATIONS & FUTURE WORK

- **Immediate:**
  1. Expand lexicographic resources for rare forms.
  2. Add syntactic parsing (dependency parser).
  3. Integrate contextualised embeddings (Tajik BERT).
  4. Build supervised ML models for classification.
- **Long-term:**
  1. Dedicated TajikPersian transliteration module.
  2. Trainable NER using toponym and personal name datasets.
  3. Abstractive and extractive summarisation.
  4. Continuous expansion of datasets (dialectal/colloquial).
  5. Support endangered Pamiri languages: Yazghulami, Shughni, Wakhi, Ishkashimi, Sariqoli, Rushani, Bartangi, Khufi, Oroshori.

TEST COVERAGE (616 tests, 93% overall)

Module	Tests	Cov. (%)
Preprocessing	33	100
Tokenization	22	88
Subword Tokenization	12	100
Sentence Splitting	7	93
POS Tagging	9	87
Morpheme Segmentation	8	98
Lemmatization	23	92
Stemming	6	87
Stop Words	6	95
NER / Alignment	5	84
Script Detection	47	95
Validation & Metrics	18	99
Embeddings	15	83
Feature Extraction	13	96
Keyword Classification	20	98
Unified Morphology	13	93
Sentiment Analysis	15	94
Core & Pipeline	39	99
Config. & Errors	23	85
Utilities & Resources	26	100

CONCLUSION & HOW TO CITE

- TajikNLP is the first holistic NLP system for Tajik Cyrillic.
- Open-source (MIT), available on PyPI and GitHub.
- Combines classical linguistic methods with modern neural components.
- Unified morphology engine improves handling of agglutinative inflections.
- Four openly released datasets (Apache 2.0) support further research.

Cite: *Arabov, M., Habibullozoda, K., & Shirinov, N. (2026). TajikNLP: An Open-Source Toolkit for Comprehensive Text Processing of Tajik (Cyrillic Script). In Proceedings of CLIB 2026.*

ACKNOWLEDGEMENTS

The authors thank the open-source community and all contributors.