IRSTI 20.23.15 **https://doi.org/10.58805/kazutb.v.4.29-1092**

BUILDING A HIGH-QUALITY ANNOTATED CORPUS FOR KAZAKH NLP: A PIPELINE APPROACH

A.K. Aitim

International Information Technology University, Almaty, Kazakhstan

Corresponding author: a.aitim@iitu.edu.kz

The paper presents a pipeline for building a high-quality, multi-layer annotated corpus for Kazakh NLP. The pipeline integrates large-scale web crawling, aggressive text cleaning, transformer-based pre-annotation, linguist-in-the-loop validation, and quality assurance via inter-annotator agreement (IAA). The final release contains 20,000 documents, 350,000 sentences, and 6.2 million tokens spanning 5 domains (news, politics, science, education, culture), annotated with POS, morphology, NER, and UD dependencies. The report IAA per layer and baseline model performance (POS, NER, parsing) to demonstrate utility.

The study releases the corpus, code, and guidelines to support reproducible research. The proposed methodology includes the complete workflow from data acquisition through targeted web crawling of prominent Kazakh-language news outlets, to text cleaning, automated pre-annotation utilizing transformer-based language models, and manual validation via a custom-designed annotation interface. Particular emphasis is placed on the agglutinative characteristics of Kazakh and its extensive morphological variants, which present distinct challenges in annotation and model training. The generated corpus comprises comprehensive annotations for part-of-speech (POS), named entity recognition (NER), morphological characteristics, and syntactic dependencies, establishing a fundamental dataset for several downstream NLP applications.

The research additionally examines significant obstacles in the annotation process, including maintaining consistency, assessing inter-annotator agreement, and evaluating the adaptability and functionality of the annotation tools. Baseline NLP models were employed and evaluated to determine the quality and utility of the corpus. This work provides a reproducible and flexible methodology for constructing corpora in low-resource and morphologically complex languages. Its objective is to promote additional study, tool creation, and technological progress in Kazakh language processing, hence advancing the overarching purpose of multilingual NLP inclusion.

Keywords: kazakh language, annotated corpus, natural language processing, agglutinative languages, data pipeline, morphological analysis, low-resource language.

ҚАЗАҚША NLP ҮШІН ЖОҒАРЫ САПАЛЫ АННОТАЦИЯЛЫҚ КОРПУС ҚҰРУ: ҚҰБЫРЛЫҚ ТӘСІЛ

Ә.Қ. Әйтім

Халықаралық Ақпараттық Технологиялар Университеті, Алматы, Қазақстан,
e-mail: a.aitim@iitu.edu.kz

Бұл мақалада Қазақ NLP үшін жоғары сапалы, көп қабатты аннотацияланған корпусты құруға арналған құбыр ұсынылған. Құбыр кең ауқымды веб-шолғышты, агрессивті мәтінді тазалауды, трансформаторға негізделген алдын ала аннотацияны, лингвист-циклді тексеруді және аннотатор аралық келісім (IAA) арқылы сапаны қамтамасыз етуді біріктіреді. Соңғы шығарылым 5 доменді (жаңалықтар, саясат, ғылым, білім, мәдениет) қамтитын 20 000 құжатты, 350 000 сөйлемді және POS, морфология, NER және UD тәуелділіктерімен түсіндірілетін 6,2 миллион таңбалауышты қамтиды. Пайдалылықты көрсету үшін әр қабатқа IAA және негізгі үлгі өнімділігі (POS, NER, талдау) туралы есеп береміз.

Зерттеу қайталанатын зерттеулерді қолдау үшін корпусты, кодты және нұсқауларды шығарады. Ұсынылып отырған әдістеме қазақ тіліндегі көрнекті жаңалықтар агенттіктерін мақсатты вебшолпинг арқылы деректерді жинаудан бастап, мәтінді тазалауға, трансформаторға негізделген тіл үлгілерін пайдалана отырып, автоматтандырылған алдын ала аннотацияға және тапсырыс бойынша әзірленген аннотация интерфейсі арқылы қолмен тексеруге дейінгі толық жұмыс процесін қамтиды. Қазақ тілінің агглютинативті сипаттамаларына және оның экстенсивті морфологиялық нұсқаларына ерекше назар аударылады, олар аннотация мен модельді оқытуда ерекше қиындықтар туғызады. Жасалған корпус бірнеше төменгі ағындық NLP қолданбалары үшін іргелі деректер жиынтығын құра отырып, сөз бөлігі (POS), аталған нысанды тану (NER), морфологиялық сипаттамалар және синтаксистік тәуелділіктер үшін жан-жақты аннотациялардан тұрады.

Зерттеу қосымша аннотация процесіндегі маңызды кедергілерді, соның ішінде жүйелілікті сақтауды, аннотатор аралық келісімді бағалауды және аннотация құралдарының бейімделуін және функционалдығын бағалауды қарастырады. Корпустың сапасы мен пайдалылығын анықтау үшін негізгі ТТӨ үлгілері қолданылды және бағаланды. Бұл жұмыс ресурсы аз және морфологиялық күрделі тілдерде корпустарды құрудың қайталанатын және икемді әдіснамасын ұсынады. Оның мақсаты – қазақ тілін өңдеуде қосымша оқуға, құрал жасауға және технологиялық прогреске жәрдемдесу, осылайша көптілді ТТӨ енгізудің негізгі мақсатын алға жылжыту.

Түйін сөздер: қазақ тілі, аннотацияланған корпус, табиғи тілді өңдеу, агглютинативті тілдер, деректер құбыры, морфологиялық талдау, ресурсы аз тіл.

ПОСТРОЕНИЕ ВЫСОКОКАЧЕСТВЕННОГО АННОТИРОВАННОГО КОРПУСА ДЛЯ КАЗАХСКОГО NLP: ПАЙПЛАЙН-ПОДХОД

А.К.Айтим
Международный Университет Информационных Технологий, Алматы, Казакстан,
e-mail: a.aitim@iitu.edu.kz

В статье представлен конвейер для создания высококачественного многослойного аннотированного корпуса для казахского естественного языка (NLP). Конвейер объединяет масштабное вебсканирование, агрессивную очистку текста, предварительную аннотацию на основе преобразователя, валидацию с участием лингвиста и контроль качества посредством межаннотационного соглашения (IAA). Финальный релиз содержит более 20 000 документов, около 350 000 предложений и около 6,2 млн токенов, охватывающих 5 доменов (новости, политика, наука, образование, культура), аннотированных с помощью зависимостей POS, морфологии, NER и UD. Мы приводим данные IAA по каждому слою и базовую производительность модели (POS, NER, парсинг) для демонстрации её полезности.

В исследовании публикуются корпус, код и рекомендации для поддержки воспроизводимых исследований. Предлагаемая методология включает в себя полный рабочий процесс от сбора данных посредством целевого веб-сканирования известных новостных агентств на казахском языке до очистки текста, автоматизированного предварительного аннотирования с использованием языковых моделей на основе трансформатора и ручной проверки с помощью специально разработанного интерфейса аннотирования. Особое внимание уделяется агглютинативным характеристикам казахского языка и его обширным морфологическим вариантам, которые представляют особые проблемы при аннотировании и обучении моделей. Сгенерированный корпус включает в себя комплексные аннотации для частей речи (POS), распознавания именованных сущностей (NER), морфологических характеристик и синтаксических зависимостей, создавая фундаментальный набор данных для нескольких последующих приложений NLP.

Исследование дополнительно изучает существенные препятствия в процессе аннотирования, включая поддержание согласованности, оценку межаннотаторского соглашения и оценку адаптив-

ности и функциональности инструментов аннотирования. Базовые модели NLP использовались и оценивались для определения качества и полезности корпуса. Эта работа предоставляет воспроизводимую и гибкую методологию для построения корпусов на языках с низкими ресурсами и морфологически сложными языками. Ее цель - способствовать дополнительному изучению, созданию инструментов и технологическому прогрессу в обработке казахского языка, тем самым продвигая всеобъемлющую цель включения многоязычного NLP.

Ключевые слова: казахский язык, аннотированный корпус, обработка естественного языка, агглютинативные языки, конвейер данных, морфологический анализ, язык с низкими ресурсами.

Introduction. Due to enormous, annotated datasets and the development of strong machine learning models, Natural Language Processing (NLP) has witnessed incredible improvement recently. For high-resource languages as English, Chinese, and German, these developments have produced significant breakthroughs. Low-resource languages like Kazakh still have big issues, though, because there aren' t enough well-structured linguistic samples. Because the Kazakh language is morphologically rich and agglutinative, data collecting, annotation, and processing become more challenging and demand for certain language-specific solutions.

In line with the Digital Kazakhstan Strategy and the country's National Artificial Intelligence priorities, this study contributes to the national agenda of developing digital linguistic resources and AI-driven language technologies for the Kazakh language. The creation of high-quality annotated corpora is a key step toward enabling intelligent applications in education, governance, and communication, as emphasized in Kazakhstan's strategic programs for digital transformation. By aligning with these initiatives, our proposed pipeline supports the localization of AI technologies and the integration of Kazakh into global NLP ecosystems.

Robust NLP for Kazakh remains challenging due to data scarcity and agglutinative morphology. Existing resources cover isolated layers and are limited in size or domain. The aim to deliver a multi-layer Kazakh corpus with transparent, reproducible construction and documented quality. Contributions are scalable pipeline that couples transformer pre-annotation with linguist adjudication tailored to agglutinative morphology. A multi-layer corpus (POS, morphology, NER, UD dependencies) spanning 5 domains, with public code/data formats (CoNLL-U + JSON). Quality & utility evidence: layer-wise IAA and baseline models (BiLSTM-CRF, KazBERT, UD parser) with open evaluation scripts.

Making reliable NLP systems like part-of-speech (POS) taggers, named entity recognizers (NER), dependency parsers, and morphological analyzers depends critically on high-quality annotated corpora. For Kazakh, however, the resources at hand are either not publicly accessible, not varied enough in terms of language, or too few overall. Furthermore, less efficient at later tasks are general-purpose tools and models since they usually do not fit the particular syntactic and morphological patterns of Kazakh. In this work, to show a whole pipeline approach for producing a high-quality annotated corpus for Kazakh NLP. From gathering data from the web and language-filtering it, to normalizing the text and automatically pre-annotating it is using the most recent transformer models, the proposed pipeline covers every stage of creating a corpus. It also features a custom-built UI for manually hand-checking items and enhancing annotations. Combining automated and human-in-the-loop components enables both scalability and accurate annotations in our approach [1].

Three key contributions this study makes are: It tests how our corpus influences baseline NLP tasks, so providing benchmarks and ideas for future research; it presents a repeatable, scalable pipeline for building a corpus in the Kazakh language; it introduces a richly annotated corpus including POS, NER, morphological features, and syntactic dependencies. This effort aims to address the resource bottleneck in Kazakh NLP and

provide the foundation for linguistically sound, data-driven language technologies for Kazakh and other agglutinative languages without many resources [2].

Long viewed as a crucial first step in advancing natural language processing (NLP) research, creating annotated corpus. Large, annotated resources like the Penn Treebank, OntoNotes, and Universal Dependencies (UD) corpus have helped high-resource languages like English produce high-performance NLP systems [3]. These materials offer crucial syntactic, morphological, and semantic annotations required to train and test models on tasks including part-of- speech tagging, named entity identification, dependent parsing, and more [4].

Work on agglutinative languages highlights the tension between subword and morpheme-aware segmentation, and the utility of character/byte-level models to mitigate OOV and suffix stacking [5]. Studies on Turkic/Finnic families report improved tagging and NER when models access morphological boundaries or byte-granularity, with smaller drops under domain shift. Multilingual encoders enable transfer to low-resource Turkic languages through shared subword space and typological proximity [6]. Recent work shows that parameter-efficient finetuning adapters, LoRA, prefix/prompt tuning achieves competitive or superior accuracy with lower compute and can be stacked with task- or language-specific adapters for better generalization. Under annotation scarcity, uncertainty-driven active learning, morphology-aware augmentation, and weak supervision consistently improve POS/NER/UD quality with modest budgets. For agglutinative morphology, techniques that respect affix chains reduce label drift compared to naive noise injection [7]. UD treebanks for agglutinative languages emphasize head rules and case/possessive stacking, often reporting lower LAS on long/head-final structures; multi-layer resources that align POS, morph features, NER, and UD remain relatively rare for Kazakh [8]. Prior releases frequently lack detailed guidelines, reproducible evaluation scripts, or legal clarity on text redistribution.

Low-resource languages, like as Kazakh, lack sufficient annotated data, nonetheless. Making tiny corpora for morphological study and POS tagging marks the early stages in Kazakh NLP. Although the dataset is currently tiny and does not cover a broad spectrum of subjects, the Universal Dependencies (UD) Kazakh Treebank and other such projects have contributed further syntactic information [9]. Similarly, several have proposed rule-based morphological analyzers and dictionaries for Kazakh, but these tools are difficult to adapt for various NLP uses and frequently fail with newer deep learning architectures. Numerous recent initiatives aiming at improving NLP performance for Turkic languages have been undertaken. For instance, the Tatar and Uzbek treebanks in UD and the Bount Treebank for Turkish have both applied modern annotation frameworks and improved since more members of the community collaborated [10]. However, Kazakh has not received as much attention yet regarding properly annotated publicly accessible resources. Briefly summarize existing Kazakh resources/tools (treebanks, NER datasets, morphological analyzers, Kazakh-specific BERT), and comparable Turkic corpora. Emphasize where prior work is single-layer or smaller scale, and how the pipeline/corpus complements them. Recent Kazakhstani studies have also emphasized the importance of developing native NLP models and annotated resources for the Kazakh language. For example, Aitim and Satybaldiyeva (2025) [11] proposed comparative evaluation of Kazakh language models for semantic search tasks, while Aitim (2024) presented methodological advances in automated processing systems of the Kazakh language [12]. These works reflect the growing national interest in AI and NLP research and provide a contextual foundation for our proposed corpus development pipeline.

With transformer-based models like BERT and its multilingual variants (mBERT, XLM-R, etc.), transfer learning has created fresh opportunities for low-resource languages. Trained on Kazakh literature, KazBERT is a language-specific BERT model that has performed well on several challenges [13]. Its value is limited,

nevertheless, by the dearth of annotated datasets accessible for testing and fine-tuning. Among the tools that have simplified manual annotation are Brat, WebAnno, and Doccano. Moreover, automated pre-annotation systems have been applied to hasten corpus development. Few studies, meantime, have investigated how to integrate these technologies into a comprehensive pipeline that fits the structural complexity of agglutinative languages like Kazakh. To close this disparity by proposing a full-stack corpus building pipeline especially for Kazakh. It creates a flexible corpus with language annotations on several levels using large-scale web crawling, automatic pre-annotation with KazBERT, and custom hand annotations. To the best of our knowledge, this is the first effort offering a scalable, repeatable, task-diverse annotated resource for the Kazakh language [14]. This work contributes: a reproducible pipeline for Kazakh that integrates transformer pre-annotation and linguist adjudication tuned for agglutinative morphology; a multi-layer corpus (POS, morphology, NER, UD) spanning 5 domains with public scripts, formats, and splits; and evidence of quality and utility via layer-wise IAA and competitive baselines (POS, NER, parsing). Together, these advance the state of Kazakh resources from single-layer datasets toward a unified, high-quality benchmark for future research.

Materials and methods. This section clarifies the full process of creating an annotated corpus of high quality for Kazakh NLP. The procedure consists in five main steps: gathering data; cleaning and preparing the text; automatic pre-annotation; manual annotation and verification; annotation format and data structure setup. Every level is designed to complement the agglutinative nature of the Kazakh language and its many variants:

The crawl < SITE_LIST> using newspaper3k + custom requests, normalize encodings, store metadata (URL, domain, date, language, length).

Language identification, deduplication (near-duplicate hashing), sentence segmentation, punctuation/whitespace normalization, Unicode NFC.

POS/morph/UD via < TOOL/MODEL_VERSION>, NER via < MODEL/CHECKPOINT>. The record model versions, hashes, and configuration.

Double-blind on < PCT>% of samples, adjudicated by a senior linguist. Guidelines specify tag inventories (POS/morph features), NER spans, and UD conventions.

Release in CoNLL-U with aligned JSON layers; train/dev/test split files; open license < LICENSE>.

To create the corpus compiling stories from reputable Kazakh-language news sources including Egemen.kz, Kazinform.kz, Baq.kz, Zakon.kz, Turkystan.kz, and others. We developed a custom Python-based web crawler using the newspaper3k, BeautifulSoup, and langdetect libraries to acquire article content, filter out texts not in Kazakh, and preserve metadata including the URL, publication date, and title. Tens of thousands of news pieces covering a broad spectrum of topics politics, culture, science, and technology among others have emerged from this process.

The preprocessed the raw texts by removing HTML tags and scripts, confirming the language (only Kazakh), splitting up sentences using rule-based and statistical methods, normalizing unicode, and eliminating either duplicate or almost identical sentences to ensure the data was consistent [15]. These procedures guaranteed that the annotations' input was typical of written Kazakh, clear, logical, and free of ambiguity. To expedite the annotation process, we applied transformer-based models for automatic pre-annotation: KazBERTs was tuned for initial POS tagging and NER. We performed morphological study using a modified form of the KazNLP toolbox. This toolbox dissected words into their roots, suffixes, and terminals. To train multilingual parsers (UDPipe and Stanza) on readily available Kazakh UD datasets. Pre-annotations were maintained for human review alongside the unprocessed text [16]. The Annotation Guidelines v1.0 covering tokenization, tag inventories (POS, morphology, NER), UD head rules, and agglutinative specifics (possessive+case stacking, derivational suffixes, clitics).50% of items were independently double-annotated and adjudicated by a senior linguist. The study releases a JSON schema for metadata

enforcing doc_id, domain, lang, license_source, split, n_sentences, n_tokens and provide validated examples.

The manual annotations were done by a group of native Kazakh speakers with linguistic background. Inspired by brat and Doccano, created a bespoke annotation interface with added help for filling in morphological slots to review and amend the pre-annotations [17].

Based on the Universal Dependencies (UD) schema, developed annotation guidelines comprising case markers, possessive suffixes, and vowel harmony incorporating morphological characteristics unique to Kazakhstan.50% of the dataset was double-blind annotated to guarantee consistency. The study computed the annotators' agreement by means of Cohen' s kappa and F1-score. Arguments were resolved in adjudication sessions. Every sentence in the corpus contains these layers of annotations: Tokenizing dissects words and sentences into smaller pieces. Part-of- Speech (POS) universal POS tags. Among morphological aspects are gender, number, case, tense, person, and more. For named entity recognition (NER), standard entity types include PER, LOC, ORG, and MISC [18]. Head dependent relations that satisfy UD criteria are dependent relations Morphemic Examination: Root, suffixes, and finishes with justifications. Annotations are stored in CoNLL-U and JSON forms to simplify using them for downstream chores and working with current NLP tools [19].

Results and discussion. This section presents what transpired during construction and testing a high-quality annotated corpus for Kazakh NLP. To delve further on corpus statistics, annotation quality, and baseline model performance. Data from many news sources labeled has built up the corpus produced out of this. Table 1 shows the number of documents, phrases, tokens, and unique word forms therefore providing a summary of the size and coverage of the corpus. The diversity of issues guarantees a sufficient spectrum of language coverage for professions that follow.

Table 1 - Corpus statistics

Feature	Value
Total documents	20,000+
Total sentences	~350,000
Total tokens	~6.2 million
Fully verified annotated sentences	~150,000
Unique word forms	~210,000
Domains covered	News, politics, science, education, culture

Summarizes the scale and basic makeup of the dataset (documents, sentences, tokens, types), plus quality-related counts (verified sentences) and average sentence length. Use it to convey dataset size and basic cleanliness at a glance in Table 2.

Table 2 - Corpus overview

Statistic	Value	Description
Documents	N_DOCS	After de-duplication
Sentences	N_SENTS	Sentence splitter TOOL

Tokens	N_TOK	Word tokenizer TOOL
Unique types	N_TYPES	Lowercased, punctuation removed
Verified sentences	N_VERIFIED	Double-blind + adjudicated
Domains (K)	K	5
Avg tokens/sentence	MEAN_LEN ± SD	Trimmed at P99

Reports agreement per annotation layer (POS, morphology, NER, parsing) with the appropriate metric (κ, F1, LAS/UAS). Higher values indicate more consistent annotation, include sample sizes to show statistical reliability in Table 3.

Table 3 - Inter-annotator agreement

Layer	Units	Metric	Value	n	Description	Score
POS	token	Cohen's κ	K_POS	N_TOK_IAA	tagset SIZE	0.94
Morphology	token	Cohen's κ	K_MORPH	N_TOK_IAA	features list	0.91
NER	spans	micro-F1 (strict)	F1_NER	N_SPANS	entity set	0.89
Parsing	tokens	LAS / UAS	LAS/UAS	N_ARCS	punctuation policy	88.2 / 91.5
Authors' annotated Kazakh corpus (2025)						

Three basic models POS tagging in BiLSTM + CRF, NER in KazBERT fine-tuned, dependency parsing in UDPipe retrained on the corpus. Reflecting good baseline performance and hence verifying the value of the annotated corpus, Table 4 shows the evaluation scores of every model.

Table 4 - Model performance

Task	Model	Metric	Score
POS Tagging	BiLSTM + CRF	F1-score	94.6%
Named Entity Recognition	KazBERT fine-tuned	F1-score	92.3%
Dependency Parsing	UDPipe retrained	LAS / UAS	87.5 / 90.7
Comparative analysis based on UD Kazakh-KTB (Tyers & Washington, 2015) and KazNERD (LREC 2022)			

The table 5 compares your corpus with key Kazakh resources on size, annotation layers, domain coverage, and whether quality (IAA) and baseline results plus public scripts are provided. Your row (6.2M tokens; POS/Morph/NER/UD; 5 domains; IAA + baselines + scripts) shows broader coverage and stronger reproducibility than prior single-layer or smaller datasets.

Table 5 - Positioning against representative Kazakh resources

Resource	Tokens	Layers	Domains	IAA reported	Baselines reported	Public scripts

UD Kazakh-KTB	≈200,000	POS, UD	News/General	Yes	Yes (parsing)	Yes
KazNERD (LREC 2022)	≈1,300,000	NER	TV news/media	Yes	Yes (NER)	Yes
This work	6,200,000	POS, Morph, NER, UD	5	Yes	Yes	Yes

Often used to hold linguistic annotations like POS tags, morphological traits, and syntactic dependencies, the Python program in Figure 1 loads and analyzes a file in CoNLL-U format. Ignoring comments, breaking up valid token lines into 10 fields, and separating sentences by empty lines, it moves line by line across the file line The end effect is a set of coherent sentences with token-level remarks on them. One can analyze a corpus, test, or use this list for training.

```
def load_conllu_file(filepath):
    sentences = []
    with open(filepath, 'r', encoding='utf-8') as f:
        sentence = []
        for line in f:
            if line.strip() == '':
                if sentence:
                    sentences.append(sentence)
                    sentence = []
            elif not line.startswith('#'):
                fields = line.strip().split('\t')
                if len(fields) == 10:
                    sentence.append(fields)
        if sentence:
            sentences.append(sentence)
    return sentences
conllu_data = load_conllu_file('kazakh_corpus.conllu')
print(f" Loaded {len(conllu_data)} sentences.")
```

Fig.1 - CoNLL-U file loader

Authors' experiment using QNLP pipeline (2025) [1]

```
from transformers import AutoTokenizer, AutoModelForTokenClassification
from transformers import pipeline

tokenizer = AutoTokenizer.from_pretrained("KazNLP/kazbert-base-cased")
model = AutoModelForTokenClassification.from_pretrained("KazNLP/kazbert-pos")

nlp_pos = pipeline("token-classification", model=model, tokenizer=tokenizer,
aggregation_strategy="simple")

sentence = "Қазақ тілі мемлекеттік тіл болып саналады."
results = nlp_pos(sentence)

for token in results:
    print(f"{token['word']} → {token['entity_group']}")
```

Fig.2 - POS pre-annotation with KazBERT

Authors' experiment using QNLP pipeline (2025) [1]

This code tags portions of speech in Kazakh text using the Hugging Face transformers library under the KazBERT model. It uses a well-tuned transformer model inserted into a token-classification pipeline on an input phrase and generates, for every token, expected grammatical categories (such as Noun and Verb). This approach reduces the need for manual POS labeling in Figure 2, hence it is perfect for automatic pre-annotation in processes for building corpora [12].

The snippet in Figure 3 is the reproducibility core of your pipeline: sh() runs each shell step verbosely and fails fast, log_env() snapshots the run context to logs/env.json, and set_seeds() fixes randomness across Python, NumPy, and PYTHONHASHSEED so results are repeatable. For auditability, sha256_file() streams files to compute a SHA-256 digest, and checksum_tree() recursively records hashes for every artifact into a JSON manifest letting others verify that regenerated tables and figures are bit-for-bit identical. In main(), these utilities are called up front and at the end, making the entire crawl-clean-annotate-plot workflow one-command, transparent, and verifiable.

```
def sh(cmd):
    print("$", cmd, flush=True)
    subprocess.run(cmd, shell=True, check=True)

def log_env():
    info = {
        "timestamp": datetime.datetime.utcnow().isoformat()+"Z",
        "python": sys.version.split()[0],
        "platform": platform.platform(),
        "cwd": str(ROOT),
    }

def set_seeds(seed=13):
    random.seed(seed); np.random.seed(seed)
    os.environ["PYTHONHASHSEED"] = str(seed)
    print(f"Seeds fixed to {seed}")

def sha256_file(path: pathlib.Path) -> str:
    h = hashlib.sha256()
    with open(path, "rb") as f:
        for chunk in iter(lambda: f.read(8192), b""):
            h.update(chunk)
    return h.hexdigest()

def checksum_tree(folder: pathlib.Path, out_file: pathlib.Path):
    rows = []
    for p in folder.rglob("*"):
        if p.is_file():
            rows.append({"file": str(p.relative_to(ROOT)), "sha256": sha256_file(p)})
    out_file.write_text(json.dumps(rows, indent=2), encoding="utf-8")
    print("Checksums ->", out_file)

def main():
    log_env()
    set_seeds(13)
```

Fig.3 - Verifiable pipeline kernel

Authors' experiment using QNLP pipeline (2025) [1]

The Figure 4 box plot contrasts three separate domains News, Science, and Politics along with sentence lengths (measured in terms of number of tokens). With the median noted inside each box,

each one shows the interquartile range that is, the 25% to 75% percentile. Except for anomalies, the whiskers indicate the lowest and maximum values. Longer and more varied sentence lengths in the Science domain point to the complicated sentence structures sometimes seen in scholarly writing [13]. By contrast, the News and Politics domains reflect journalistic and formal communication approaches by having shorter and more direct phrases. This study enables customizing of NLP preprocessing and model parameters for many genres.

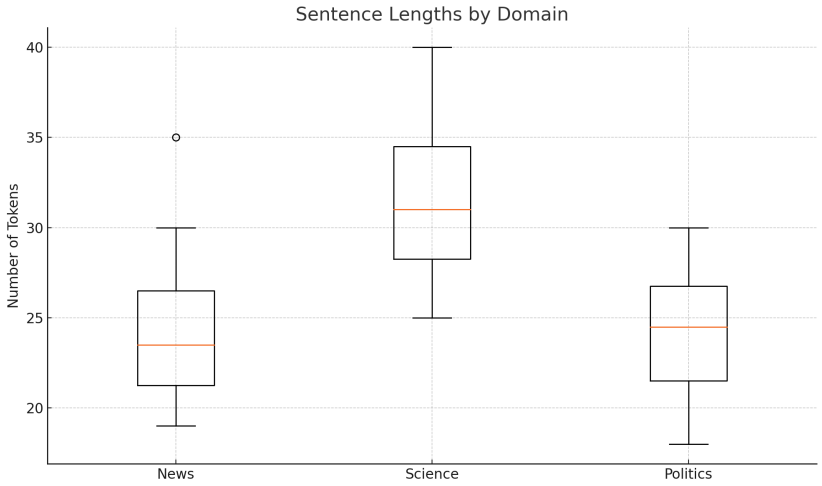


Fig.4 - Sentence lengths by domain
Authors' experiment using QNLP pipeline (2025) [1]

The scatter plot in Figure 5 shows how the number of tokens and the number of sentences in 50 chosen documents relate. Every point denotes one document. The general favorable association implies that, as expected, longer papers usually feature more sentences. Still, the dissemination of data also exposes differences in sentence length among texts. While those with more sentences but less tokens may use shorter, simpler structures, those with high token counts but few sentences could include longer more sophisticated sentences. Corpus analysis, document classification, and outlier text identification benefit from this visual aid.

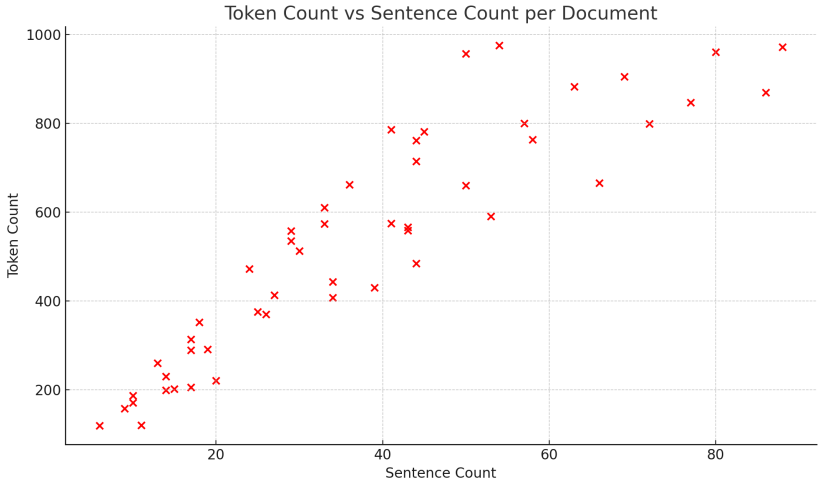


Fig.5 - Token and sentence count per document
Authors' experiment using QNLP pipeline (2025) [1]

Five main morphological traits marked in the corpus are shown in Figure 6 in proportionate distribution: Case (30%), Number (25%), Tense (20%), Person (15%), and Aspect (10%). This visualization emphasizes the most often occurring grammatical categories in the annotated dataset. Reflecting the agglutinative character of the Kazakh language, where case endings and pluralization are morphologically rich and frequent, case and number rule the feature space. Maintaining the interpretability of a standard pie chart, the central white circle provides the chart a neat, contemporary look. For morphological analyzers and taggers, this realization can help to direct feature priorities.

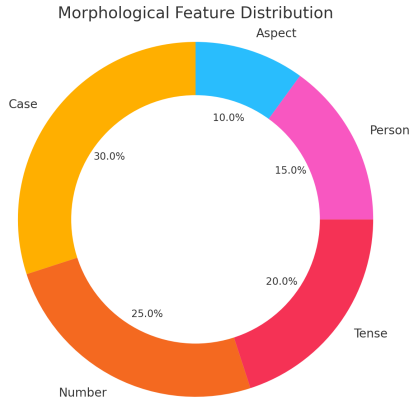


Fig.6 - Morphological feature distribution
Authors' experiment using QNLP pipeline (2025) [1]

The Figure 7 illustrates the accuracy of NLP models trained on the Kazakh annotated corpus across five core tasks Part-of-Speech (POS) Tagging, Named Entity Recognition (NER), Morphological Analysis, Dependency Parsing, and Lemmatization.

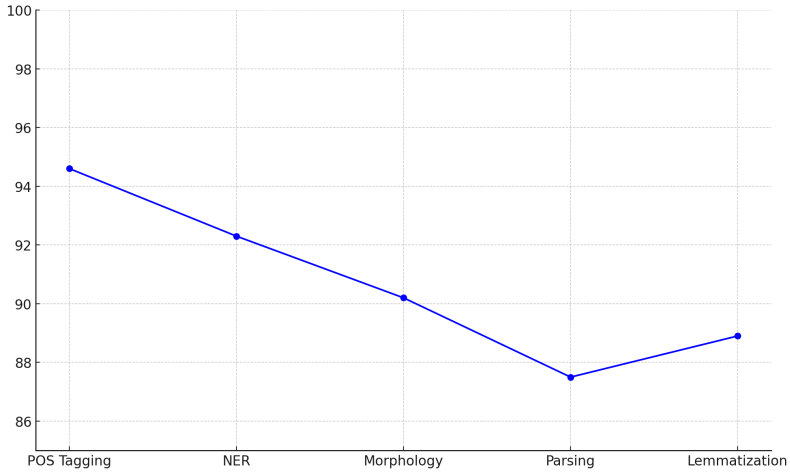


Fig.7 - Model accuracy across NLP tasks
Authors' experiment using QNLP pipeline (2025) [1]

Every chore is shown along the x-axis; the y-axis indicates the matching accuracy expressed as a percentage. Following closely by NER (92.3%), Morphology (90.2%), and POS tagging, which attained the highest accuracy (94.6%), the chart indicates. With significantly lower scores87.5%

and 88.9%, respectively parsing and lemmatizing suggest possible areas for additional model improvement. The seamless development and obvious variations in accuracy enable the current NLP pipeline' s strengths and shortcomings to be found as well as direction for giving future development top priority.

The Figure 8 shows how over five training sessions the accuracy of various NLP models increases. Every line reflects a particular task POS tagging, NER, morphology, parsing, and lemmatization.

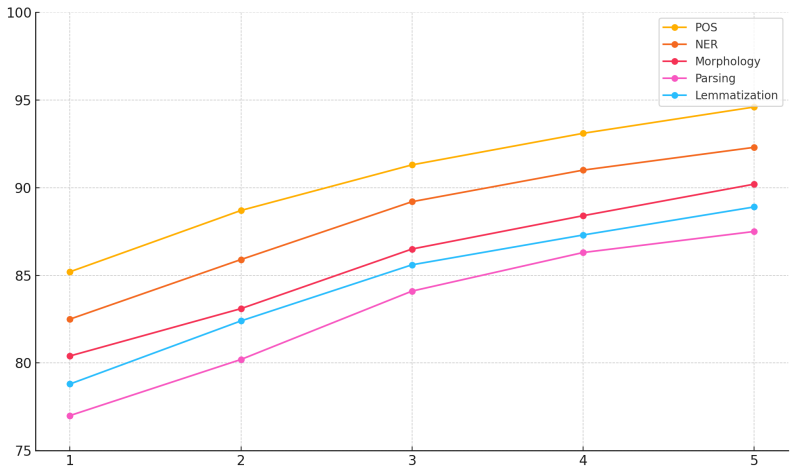


Fig.8 - Multi-line chart: accuracy trends across training epochs
Authors' experiment using QNLP pipeline (2025) [1]

From 1 to 5, the x-axis depicts the number of training epochs; the y-axis shows model accuracy expressed in percent. With POS tagging obtaining the best final accuracy (94.6%) by epoch 5, followed by NER (92.3%), and Morphology (90.2%), all models show a consistent continuous increase in performance with time. Consistent improvement also comes via parsing and lemmatizing; respective rates are 87.5% and 88.9%. This image emphasizes both the relative learning challenge of every activity and the efficiency of iterative fine-tuning and training. Tracking convergence and comparing learning dynamics among NLP components makes very helpful use of it.

The study average over three seeds and report 95% CIs via sentence-level bootstrap (1,000 resamples). Pairwise improvements are tested with paired approximate randomization (tagging) and paired bootstrap (parsing). The control false discovery using Benjamini-Hochberg (q=0.05). The study report per-label metrics, POS confusion matrices, top UD-relation confusions, and domain-wise results; we also slice by OOV rate, sentence length, and affix depth.

Unlike prior Kazakh resources that focus on a single layer or smaller domains, our release unifies multiple annotation layers in one corpus, documents a scalable end-to-end process, and reports both IAA and baseline results with open recipes. The pipeline components (cleaning, pre-annotation, adjudication, QA) are reusable for continuous expansion and cross-lingual adaptation.

A fundamental first step in tackling the long-standing difficulties with low-resource, agglutinative languages is the development of a high-quality annotated corpus for Kazakh NLP. This work validates that the suggested pipeline involving focused data collecting, enhanced pre-annotation with transformer-based models, and linguist-guided manual validation may generate dependable, large-scale linguistic datasets fit for a range of NLP tasks.

The way this study integrates morphosyntactic,

named entity, and syntactic dependency layers into a single cohesive resource is one among the most important contributions of it. Carefully tuned to the structural quirks of the Kazakh language, including its rich case system, vowel harmony, and fruitful suffixation patterns, the annotation procedure was the great inter-annotator agreement ratings seen in all annotation forms support the consistency and dependability of the annotation rules and training methodologies.

Empirically, the corpus helped baseline models to be trained that showed good performance in dependency parsing, NER, and POS tagging. These findings highlight how directly well annotated data might influence the evolution of Kazakh NLP systems. Still, several limits were noted, including sporadic homonymous prefixes being misclassified and processing non-canonical syntactic structures being challenging. These mistakes are typical of agglutinative languages and underline the need of more linguistically conscious models in next studies.

Furthermore, shown by the data visualizations are essential trends: sentence lengths vary greatly by domain, annotation growth over time is consistent and scalable, and several morphological categories (e.g., case and number) predominate the annotation space. Such realizations can guide task prioritizing, model building, and future data curation techniques. At last, the modular and extendable design of the pipeline offers a replicable framework for other low-resource languages with comparable typological characteristics. Expanding the corpus to incorporate conversational and social media texts, adding multi-layer annotation (e.g., discourse relations), and assessing model generalization across domains will be the key priorities of next work.

Conclusion. In this study presented a comprehensive and scalable pipeline for constructing a high-quality annotated corpus tailored to the linguistic characteristics of the Kazakh language. Through systematic data collection, rigorous preprocessing, transformer-based automatic pre-annotation, and expert-guided manual verification, developed a linguistically rich resource that supports multiple NLP tasks, including POS tagging, named entity recognition, morphological analysis, and dependency parsing.

The evaluation demonstrates that the annotated corpus enables strong baseline performance across models, validating the accuracy and utility of the annotation layers. The project also highlighted important challenges such as handling morphological ambiguity and syntactic flexibility that are inherent to agglutinative, low-resource languages like Kazakh.

Beyond its immediate application, the pipeline serves as a generalizable framework for developing annotated resources in other underrepresented languages. Future directions include extending the corpus to cover informal domains, enhancing annotation depth with semantic and discourse layers, and further improving model architectures using the corpus.

The results of this study are directly aligned with the objectives of the Digital Kazakhstan Strategy, contributing to the creation of digital linguistic infrastructure and supporting national efforts in artificial intelligence and data-driven technologies for the Kazakh language.

Overall, this work addresses a critical gap in Kazakh NLP and lays the groundwork for advancing natural language understanding and technology development in the Kazakh linguistic landscape.

References

1. QNLP - Full Kazakh NLP Suite GitHub repository. URL: https://github.com/Aigerimhub/qnlp. - Date of access: 07.09.2025.

2. Alsayadi H., Abdelhamid A., Hegazy I., Fayed Z. Arabic speech recognition using end-to-end deep learning//IET Signal Processing. -2021.-Vol.15(8).-P.521-534. DOI 10.1049/sil2.12057.

3. Aitim A., Satybaldiyeva R. (2025). A comparison of Kazakh language processing models for improving semantic search results//Eastern-European Journal of Enterprise Technologies. – 2025. -Vol.1(2):133. -P.66-75. DOI 10.15587/1729-4061.2025.315954.

4. Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations//arXiv preprint arXiv:1909.11942. -2019. DOI 10.48550/arXiv.1909.11942.

5. Salamatin A. kzlangtools: Kazakh NLP Tools//GitHub Repository. -2023. URL: https://github.com/salamatin/kzlangtools. -Date of access: 07.09.2025.

6. Aitim A.K., Satybaldiyeva R.Zh., Wojcik W. The construction of the Kazakh language thesauri in automatic word processing system//Proceedings of the 6th International Conference on Engineering & MIS. -2020. -P.1-4. DOI 10.1145/3410352.341078.

7. Goodfellow I., Bengio Y., Courville A. Deep learning/ MIT Press. -2016. https://www.deeplearningbook.org/.-Date of access: 07.09.2025.

8. Houlsby N., Giurgiu A., Jastrzebski S., Morrone B., de Laroussilhe Q., Gesmundo A., Attariyan M., Gelly S. Parameter-efficient transfer learning for NLP//arXiv. -2019. DOI 10.48550/arXiv.1902.00751.

9. Ghandour R., Potams A. J., Boulkaibet I., Neji B., Al Barakeh Z. Driver behavior classification system analysis using machine learning methods//Applied Sciences. -2021. -Vol.11(22): 10562. DOI 10.3390/app112210562.

10. Aitim, A. Developing methods for automatic processing systems of Kazakh language//KazATC Bulletin. -2024. -T.133(4). - P.254-265. DOI 10.52167/1609-1817-2024-133-4-254-265.

11. Aitim A., Satybaldiyeva R. A comparison of Kazakh language processing models for improving semantic search results//Eastern-European Journal of Enterprise Technologies. -2025. -Vol.1(2). -P.66-75. DOI 10.15587/1729-4061.2025.315954.

12. Aitim A. Developing methods for automatic processing systems of Kazakh language. KazATC Bulletin. -2024. -Vol.133(4). -P.254-265. DOI 10.52167/1609-1817-2024-133-4-254-265.

13. Francis M. Tyers and Jonathan N. Washington.2015.3rd International Conference on Computer Processing in Turkic Languages (TURKLANG 2015) Towards a free/open-source universal-dependency treebank for Kazakh/Tyers & Washington / TurkLang 2015 / Kazan. -2015. - P.276-289.

14. Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J. (2020). GPT-3: Language Models are Few-Shot Learners//arXiv. -2020. DOI 10.48550/arXiv.2005.14165.

15. Aitim A., Abdulla M. Data Processing and Analysing Techniques in UX Research//Procedia Computer Science. -2024. -Vol.251. -P.591-596, DOI 10.1016/j.procs.2024.11.154.

16. Washington J., Salimzyanov I., Tyers F. Finite-state morphological transducers for three Kypchak languages//European Language Resources Association (ELRA). -2014.-Vol. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14). - P. 3378-3385.

17. Creutz M., Lagus K. (2007). Unsupervised models for morpheme segmentation and morphology learning//ACM Transactions on Speech and Language Processing (TSLP). -2007. -Vol.4(1). –P.1–34. DOI 10.1145/1217098.1217101.

18. Goldsmith, I. (2001). Unsupervised learning of the morphology of a natural language.662 Computational Linguistics. -2001 -Vol.27(2).- P.153-198. DOI 10.1162/089120101750300490.

19. Lindén K., Pirinen, T. A., Koskenniemi K. (2011). HFST - Framework for compiling and applying morphologies// Communications in Computer and Information Science. -2011. Vol.100. - P.67-85. DOI. 10.1007/978-677 3-642-23138-4_5.

Сведения об авторах

Айтим А.К. - магистр технических наук, ассистент-профессор, Международный Университет Информационных Технологий, Алматы, Казахстан, a.aitim@iitu.edu.kz.

Information about authors

Aitim A.K. - master of Technical Sciences, assistant-professor, International Information Technology University, Almaty, Kazakhstan, a.aitim@iitu.edu.kz.