

# SEMANTICALLY ENHANCED MULTIMODAL NEURAL ARCHITECTURE FOR UZBEK SIGN LANGUAGE TRANSLATION

Juraev D.B, Ochilov M.M, Abdullaeva M.I.

Department of Robotics and Intellectual Systems, Tashkent University of information technologies named after Muhammad Al-Khwarizmi, Tashkent, Uzbekistan

## ABSTRACT

*This study proposes a semantically enriched multimodal neural architecture for automatic translation of Uzbek Sign Language (UZSL). In the approach, first, the text stream is brought to semantic consistency through  $n$ -gram tokenization, lemmatization, and morphological normalization, and a high-information context is prepared for BERT-MLM. Then, hand, face, and body landmarks are extracted from video frames using MediaPipe Holistic, speech transcriptions are generated using Speech-to-Text, and they are annotated with gloss dictionary-based labels in the text→gloss→landmark relationship. During the translation process, the text is fed to BERT-MLM, and semantic candidates are generated using a Top-k masking strategy; these candidates are checked against a standard gloss corpus in the constraint layer and filtered using semantic similarity calculations based on Word2Vec. Finally, a re-ranking mechanism based on SBERT evaluates the cosine similarity between the context and glosses and selects the most appropriate gloss. This integrated pipeline provides a near-real-time, consistent, and scalable translation workflow for UZSL by reducing the morphological complexity of the text, extracting stable features from multimodal signals, and selecting glosses based on semantic criteria with dictionary constraints.*

## KEYWORDS

*NLP,  $n$ -gram tokenization, lemmatization, morphological normalization, BERT-MLM, Top-k masking, gloss dictionary, dictionary constraint layer, Word2Vec, SBERT re-ranking, MediaPipe Holistic, Speech-to-Text, transformer.*

## 1. INTRODUCTION

Today, the rapid advancement of natural language processing and artificial intelligence technologies has greatly contributed to the enhancement of machine translation systems. One practical use of these technologies is the creation of sign language translation systems designed to close the communication gap between individuals with hearing or speech impairments and those without them. Although notable advancements have been achieved in widely recognized sign languages like American Sign Language and British Sign Language, research on Uzbek Sign Language (UZSL) remains scarce and underexplored. Uzbek Sign Language is a distinctive visual system that embodies the cultural and linguistic traits of Uzbek-speaking individuals with hearing and speech challenges. However, because of the scarcity of digital resources, specially collected and labeled datasets, and standardized linguistic models, automatic translation between UZSL and spoken or written Uzbek poses significant challenges. The use of semantic matching algorithms provides an effective solution to these challenges by accurately representing sign sentences, understanding context, and aligning sentence-level meanings.

This study designs the stages of developing a flexible automatic translation system for Uzbek Sign Language (UZSL) based on the integration of natural language processing (NLP) and semantic similarity algorithms. The approach combines language preprocessing, deep learning-based feature extraction, and semantic matching models to enhance sign recognition accuracy and the fluency of the translated text.

The study also highlights the relevance of developing a UZSL dataset and designing a semantic matching mechanism that can analyze the complex morphological and syntactic structures inherent in the Uzbek language. The scientific results of this work create a theoretical and practical foundation for further research in the field of processing low-resource languages. Furthermore, the research findings support the advancement of inclusive communication technologies for individuals with hearing and speech disabilities, while also enabling real-time two-way translation between sign and spoken languages.

## 2. RELATED WORKS

Now, natural language processing (NLP) technologies are essential for advancing sign language translation systems. In particular, such steps as real-time tokenization of text, semantic analysis, and matching with glossary dictionaries are the mainstay of such systems. Classical probabilistic models, including n-gram models, calculate the probabilistic connections between words in a sequence, providing a simplified but effective way to form semantic sequences. Therefore, early sign language translation systems often used text analysis steps based on n-gram models [1]. However, recent studies show that transformer-based architectures allow for a more accurate analysis of linguistic features in sign language [2].

However, for morphologically rich languages, such as Uzbek, Turkish, or Finnish, such n-gram approaches are not effective enough. Because the large number of word forms and the variability of affixes in these languages reduce the generalization ability of the model. Therefore, the division of tokens into intraword morphemes using subword tokenization (BPE, WordPiece, SentencePiece) is very important in the process of detecting sign glosses. Because sign gloss dictionaries usually contain the basic forms (lemmas) of the word. Therefore, in Uzbek sign language translation systems, dividing the incoming stream of words into subword tokens facilitates their matching to gloss level units [3].

It has been observed that the use of subword n-gram features in such neural machine translation models improves BLEU indicators by 3–7 points in low-resource languages [3]. These results are also relevant in sign translation systems, where semantically close tokens are found based on n-grams during the text-to-gloss translation process. Their correspondence with the gloss dictionary is evaluated, and each morphological unit is associated with the semantically most suitable gloss. Studies have shown that affix sequences themselves convey semantic connections, that is, new meanings emerge through the combination of affixes [4]. Similar principles are also applied in the process of translation into sign language - in this case, the model processes each morpheme not as a separate token, but as a contextual unit, which preserves the semantic integrity of the gloss sequence.

In modern approaches, n-gram models are mainly integrated with embedding methods. Typically, in classical models, word sequences are analyzed based on statistical probability, while in the embedding process, this connection is represented as a vector in semantic space [5]. This leads to a semantic deepening of the n-gram approach and allows analyzing the meaning of words not only based on the sequence, but also on the context. For example, in studies, the accuracy of gloss selection in systems based on the integration of n-grams and embeddings increased by 6–8 percent, and contextual errors decreased by up to 12 percent [6]. At the same time, it can be seen

that the speed of real-time sign language translation systems has improved by an average of 1.4 times [6].

Also, creating contextual embeddings based on unigram and bigram units allows for a deeper representation of the semantic relationship of each token with its surrounding words. In this case, n-gram data serves as additional context for interpreting the word's meaning. This approach is especially effective in sign language translation systems, since glosses are usually based not on the full form of the word, but on its semantic core - the lemma [5, 6]. Typically, the embedding model compares each token with other tokens in the context, calculates their semantic distance, and selects the closest gloss variant. As a result, the model is able to determine the original gloss without moving away from the meaning under the influence of morphological changes.

In modern language models, particularly those based on transformer architectures like BERT, the limitations of classical n-gram models have been largely overcome [7, 8]. N-gram models mainly relate the probability of each token to its predecessors, which limits the ability to cover long contexts. In the transformer architecture, the self-attention mechanism simultaneously analyzes the relationships between words throughout the text. As a result, the semantic role of each token is evaluated based on the entire context. The obtained practical results show that BERT-based models achieve an average of 15–18% higher accuracy in semantic analysis compared to n-gram approaches, and the BLEU index in long context sequences increases from 0.42 to 0.57 [9, 10].

Models based on transformer architecture also show high efficiency at the morphological segmentation stage. Contextual embedding models such as BERT and RoBERTa can more accurately distinguish semantic differences between morphological forms than classical n-gram methods [7, 9]. This is especially noticeable in morphologically rich languages such as Uzbek, Turkish, Arabic, and Finnish. Recent experimental results on English have shown that the accuracy of contextual grouping of morphological units is in the range of 87–93%, and gloss transformer models improve semantic matching in sign translation by 15–20% [11]. Since the model performs segmentation and contextual analysis together, semantic consistency between glosses is maintained, and changes in word forms do not negatively affect the overall meaning. Thus, the wide-context coverage of transformer models increases semantic accuracy in sign translation systems and ensures the natural and consistent formation of gloss sequences.

The advantage of this approach in sign language translation systems is that the gloss selection process is now implemented as a single integrated sequence of tokenization, contextual analysis, and semantic matching steps [8]. In this case, the model deeply studies the morphological changes of words, extracts their semantic core, and contextually selects the most appropriate gloss for each token. This mechanism strengthens the semantic coherence between glosses in real time, making sign language translation holistic not only at the lexical level, but also semantically and pragmatically. Research results show that in models based on such a sequence, the gloss selection accuracy is more than 90%, and the semantic continuity reaches 0.92; this is an average improvement of 20% compared to classical sequential step-by-step systems [7, 12].

Methods for representing text in vector space — Word2Vec, FastText, BERT, SBERT, etc. — are the main tools for determining semantic proximity between words in natural language processing [5, 13]. These methods form the basis of the gloss selection module in sign language translation systems, since the embeddings of glosses and the embeddings of the input text are located in the same semantic space. Therefore, the system determines the gloss that best suits the context, increasing the naturalness and semantic accuracy of the translation. Practical tests have shown that the gloss-text matching accuracy with the Word2Vec model is on average 81%, and with FastText based on subword analysis, this indicator increases to 88%. Models trained on the basis

of BERT improve gloss matching up to 93%, taking into account contextual semantics; SBERT achieved 94–95% accuracy using a bidirectional embedding approach [13, 14].

Modern research indicates that the success of sign language translation systems depends on effectively combining several key stages. These steps include n-gram-based tokenization and modeling, morphological normalization, lemma segmentation, contextual embedding, and semantic matching based on a transformer architecture [15, 2]. As a result of their harmonious operation, the system translates the incoming text into glosses in a semantically holistic way, that is, it allows each unit to be interpreted correctly in its context. As a result of this approach, the semantic connection between glosses improves, and the naturalness, accuracy, and continuity of the translation increase. In studies, such integrated approaches have shown an average improvement of 10–15% in the BLEU indicator and a 12% increase in the semantic compatibility index [16].

In particular, in the new generation of translation models being developed for Uzbek Sign Language (UZSL), a multi-layered architecture is of great importance. This architecture allows for real-time analysis accuracy, improved semantic compatibility with gloss dictionaries, automatic learning of new units, and context-based differentiation of synonymous glosses. Practical results show that ASL translation systems based on contextual models such as BERT and SBERT achieve 91% accuracy in identifying semantic synonyms between glosses and 88% accuracy in distinguishing antonyms [17, 18, 12]. Transformer based models also maintain pragmatic coherence between text and gloss sequences, which ensures the overall naturalness of the translation process [2]. Thus, multi-layer integrated systems are emerging as a promising solution for automatic analysis of Uzbek sign language, semantically based gloss detection, and real-time translation quality improvement.

Thus, the combined use of vector-space embeddings, morphological normalization, and transformer-based contextual analysis in sign language translation systems significantly enhances semantic alignment, translation accuracy, and the system's learning capability. This approach represents important research directions for Uzbek Sign Language (UZSL) in automatic gloss generation, semantic adaptation, and real-time sign language translation.

### **3. METHODOLOGY**

This section describes the methodology of the sign language translation system. The proposed approach consists of three main stages. These are natural language processing, data set generation, and a neural translation model.

- In the first stage, a multimodal dataset consisting of glosses formed from pairs of sign, speech, and text is created using MediaPipe and Speech-to-Text systems.
- In the second stage, the incoming text stream is transformed into a semantically related form through tokenization, n-gram modeling, and morphological normalization.
- In the third stage, translation from text to sign sentences is performed using a neural translation network based on deep learning models such as BERT, Masking, and SBERT.

#### **3.1. Multimodal Dataset Generation Stage**

A sign language translation system is developed using methods rooted in machine translation and natural language processing. In such a system, the data stream is formed in the form of a multimodal dataset that includes not only video, but also speech, text, and gloss forms. This dataset serves as a basis for training the model, improving translation accuracy, and selecting

semantically correct sign glosses. At the stage of creating sign actions, feature extraction using the MediaPipe library plays an important role. The Holistic Landmarker module of the MediaPipe library extracts hand, face, and body landmarks from video image segments and creates a complete landmark (coordinate points) model of the human body. This model represents gestures, body positions, and facial expressions as a whole system. At the same time, speech is converted into text using the speech-to-text system. As a result, a single interconnected dataset of gesture, voice, and text components and their glosses labeled by special experts is formed.

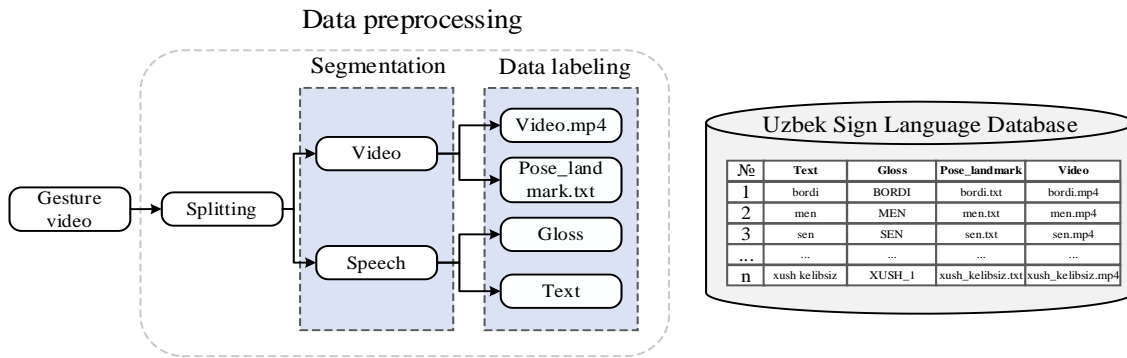


Figure 1. The process of forming a multimodal dataset of Uzbek sign language

The multimodal dataset formed in this way is the main raw material for training a sign language translation system. Data collection process consists of following stages:

1. A gesture and the corresponding speech are recorded by a camera.
2. The incoming stream is pre-processed and separated into video and speech streams.
3. Hand, face, and body landmarks are identified from the video segments using MediaPipe.
4. Text transcription is generated from the speech segments using a speech-to-text system.
5. It is labeled by experts with glosses corresponding to the sentences in the text.
6. For each gesture, video, landmark, and text data are stored in the dataset as parallel pairs (text→gloss→gesture).

The dataset is further processed using NLP technologies such as tokenization, lemmatization, n-grams, and transformer models. In this way, the system can match the semantic structure of the text with the gesture movements, allowing for natural translation in real time.

### 3.2. Natural Language Processing Step

In sign language translation, real-time natural language processing requires analyzing incoming text as a continuous stream. In such conditions, adapting the text to the sign dictionary while preserving its semantic integrity is an important scientific problem. The sequence of characters in the streamed text is first tokenized, and the language units are brought into a form ready for semantic analysis.

Tokenization is an initial but crucial stage in natural language processing, which involves dividing the sequence of characters into the smallest independent units of the language, that is, word tokens. Because a word is the minimum unit of meaning in a language. Therefore, tokenization performs not only the function of separation, but also the function of standardizing units for subsequent semantic modeling. A text string is represented as follows:

$$S = \{w_1, w_2, \dots, w_t\}$$

where  $S$  is the text,  $w_i$  are the word units. If each  $w_t$  is interpreted as a token obtained from the processing process, we represent it as follows:

$$X = \{x_1, x_2, \dots, x_i\}, \quad x_i \equiv w_t$$

where  $X$  is a sequence of tokens, and  $w$  is a dictionary of tokens. Thus, tokenization is not just a separation, but also the preparation of standardized units for further semantic modeling.

As a step after tokenization, *n-gram modeling* is used. In this method, tokens in a sequence within the text are interconnected and a statistical model is built. As a result, it becomes possible to consider not only the individual meanings of words, but also their semantic connections.

N-gram models are one of the classic approaches to studying the probabilistic connections between units in a sequence in natural language processing. Among these models, unigram, bigram, and trigram types are most commonly used.

- In the unigram model, each word is considered an independent unit, meaning that the probability of a token depends only on its frequency:

$$P(w_i),$$

- In the bigram model, the probability of a word is determined only by the word that precedes it:

$$P(w_i, w_{i+1})$$

- In the bigram model, the probability of a word is determined only by the word that precedes it:

$$P(w_i, w_{i+1}, w_{i+2}).$$

In general, the n-gram model is mathematically expressed as follows:

$$P(w_t | w_{t-1}, \dots, w_{t-n+1})$$

where  $w_t$  is the current token in the stream, and  $n$  is the length of the token units. This model allows us to evaluate relevance in the context. In this case, modeling language units through n-grams allows us to take into account their semantic and contextual connections. As a result, the system evaluates not only the probable order of arrival of the token, but also its semantic relevance.

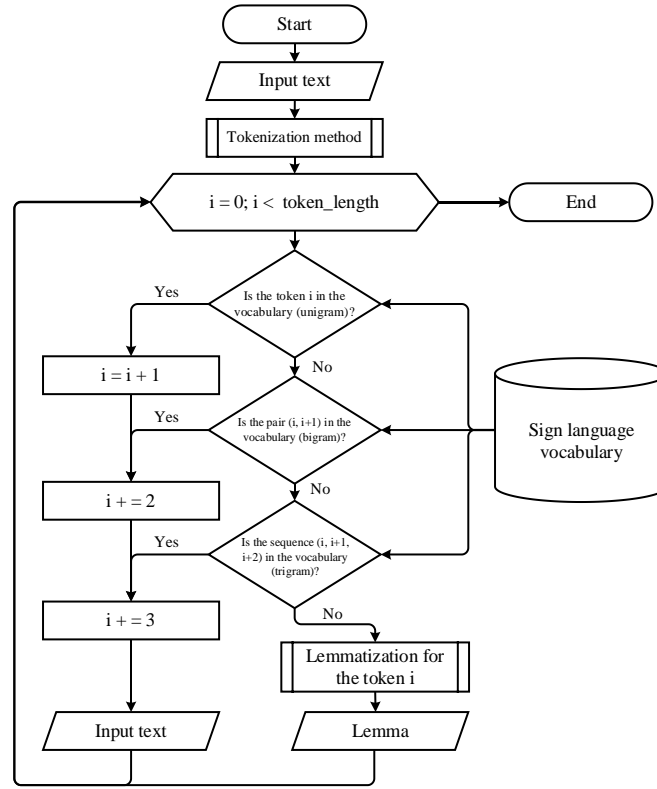


Figure 2. Algorithm for implementing N-gram tokenization

Thus, the sequence of tokens in a stream is modeled as follows:

$$S = \{w_1, w_2, \dots, w_t\}, X = \{x_1, x_2, \dots, x_i\}, x_i \equiv w_t$$

Each token is directly compared to the sign dictionary to determine the equivalent in the dictionary:

$$y = \text{Dictionary}(n\text{gramm})$$

Due to the morphological richness of natural language, units often do not appear in the dictionary form due to various affixes or grammatical changes. Therefore, morphological normalization - in particular, the process of lemmatization - is of great importance.

The process of lemmatization involves removing affixes from a word and restoring its stem form (lemma). This process restores the basic units of meaning necessary for semantic analysis. The correctness of the lemma and its correspondence to the dictionary can be determined using a probabilistic model:

$$P(r | w) = \frac{P(r | w) P(r)}{P(w)}$$

The text in the stream is semantically consistent and morphologically free, and is adapted to the sign language dictionary. This allows the translation system to accurately select glosses, combine morphological variants, and maintain semantic consistency in real time.

Thus, the n-gram modeling and lemmatization stages together form the semantic backbone of the sign language translation system. They allow for the analysis of linguistic units in a statistically optimized way, combining the formal structure of the language and the contextual meaning.

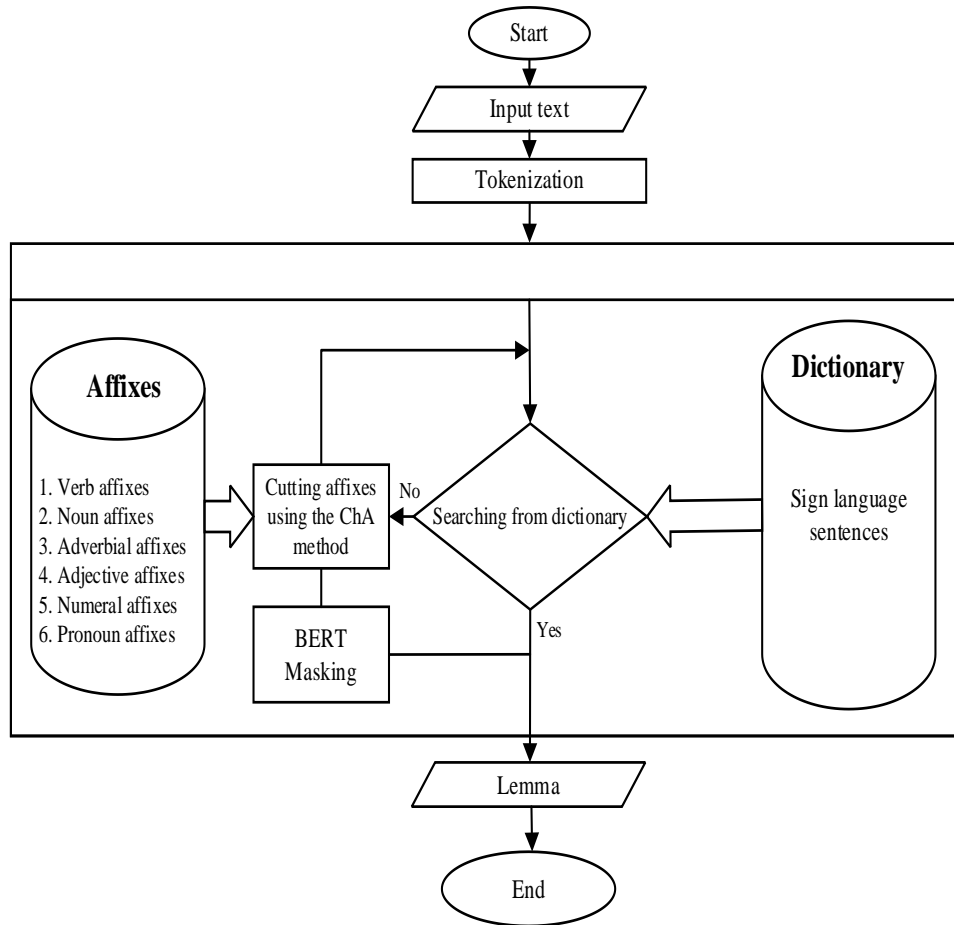


Figure 3. Block diagram of the algorithm for checking words into sign sentences using a lemma  
In the process of translation into sign language, tokenization and morphological normalization methods are of particular importance in ensuring semantic consistency. Because existing sign dictionaries usually cover only the main (lemma) units. In natural language, the text stream comes in a form enriched with various grammatical changes, additions, affixes. Therefore, it is often impossible to directly match the text in the stream with the dictionary. In this case, the n-gram tokenization model serves to identify as many direct matches as possible. Lemmatization, on the other hand, eliminates morphological changes and restores the original stem form of the word. As a result, the system brings the tokens to a semantically purified and dictionary-compatible state.

As shown in Figure 3 above, there are three main mechanisms at the heart of the process, which, working in conjunction with each other, ensure the determination of the final lemma. First, the text is divided into tokens and compared with the dictionary. If the token is in the dictionary, it is accepted as a sign equivalent. However, in many cases, the tokens differ from the forms found in the dictionary. For example, tense and person suffixes of verbs, possessive or accusative forms of nouns, and adjective suffixes distance the text from the original unity in the dictionary. In such cases, the *morphological block* comes into play. In it, word affixes are gradually separated and the root form is restored:



$$w = r + a_1 + a_2 + \dots + a_k \Rightarrow r$$

where  $r$  is the root of the word, and  $a_i$  denotes grammatical affixes. For example, the word “o‘quvchilarimizdan” (eng.from students) is shortened to the lemma “o‘quvchi” (eng.student) and this unit is matched with the dictionary. After the text has been tokenized and morphological normalized, it is necessary to identify units that are not in the dictionary.

### 3.3. Glossary Identification and Semantic Analysis Stage

If morphological analysis also fails, the process is directed to the BERT Masking model, as shown in Figure 4 below. This model is based on a transformer architecture that learns the contextual connections of words in the input sequence and predicts the most semantically appropriate variant.

In the input stage, the text is divided into tokens, which are formed from values obtained from the n-gram tokenizer. Then, the BERT model is trained using the masking method. That is, approximately 15% of the tokens in the text are selected, 80% of which are replaced with the [MASK] token. 10% are replaced with a random token, and the remaining 10% are left as is. The model learns to find hidden tokens from the context.

Then each token is converted into an embedding, that is, a digital vector. The embedding consists of the sum of three components:

- *Token embedding* - a vector representation of the word itself,
- *Position embedding* - a vector for the token's place in the sentence,
- *Segment embedding* - a vector indicating which sentence or segment it belongs to.

This process is expressed mathematically as follows:

$$x_i \rightarrow E_i = Tok\_Embedding(x_i) + Pos\_Embedding(i) + Seg\_Embedding(s)$$

where  $x_i$  is the token of order  $i$ , and  $E_i$  is its final embedding. In this way, the tokens are passed to the encoder block of the selected transformer. The resulting embeddings are then processed through *multi-headself-attention* and *feed-forward* layers. Each encoder block works as follows. Initially, the *Multi-Head Self-Attention mechanism* calculates the interconnections between the tokens:

$$Attn(Q, K, V) = Softmax\left(\frac{QK}{\sqrt{d}} + M\right)V$$

where  $Q, K, V$  represent the query, key, and value matrices. Then, semantic expressions are deepened through the *Norm* and *Feed-Forward* layers. This process is repeated 12 times in the encoder blocks, as shown in Figure 3, and each one enriches the incoming vectors with semantic information.

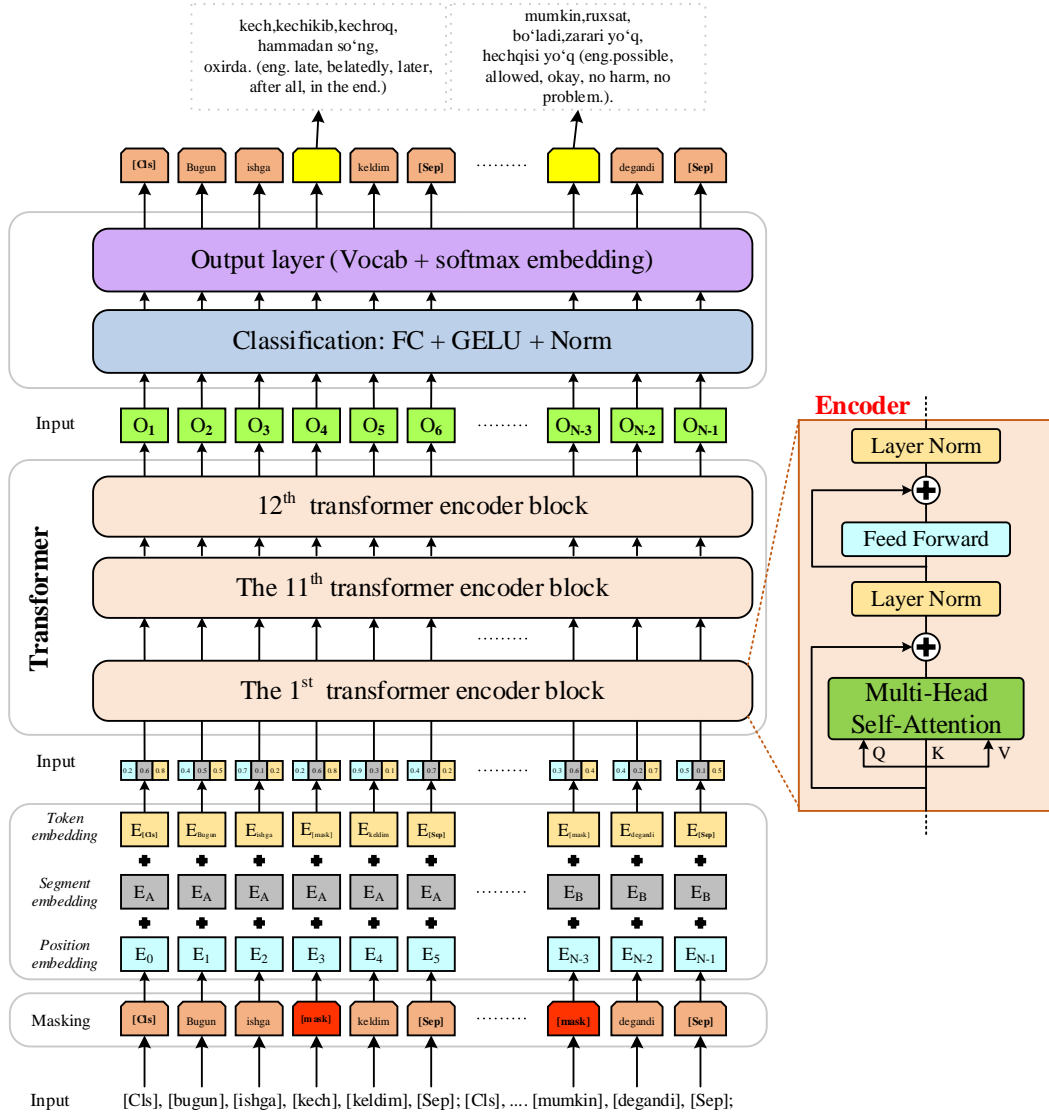


Figure 4. BERT masking architecture

At the output stage, the word vectors that have passed through the transformer encoders are sent to a special classification layer. In this layer, a fully connected neural network (FC layer), GELU activation function, and normalization processes are implemented. This stage converts the contextually enriched hidden expressions of the model into a probability distribution. In the final step, the softmax function is used to calculate the probabilities for all words in the dictionary and the unit with the highest probability is selected:

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w | X')$$

where  $X'$  is [MASK], that is, a sequence of tokens replaced by a mask, and  $V$  is a dictionary. Thus, if a token is not in the dictionary and morphological analysis also fails, it is sent to the BERT model as [MASK], and the transformer architecture predicts the most likely synonym based on the context. This prediction is associated with an equivalent in the sign dictionary.

The complexity of natural language shows that choosing only the single token with the highest probability value from the results of the BERT-based Masked Language Model often does not provide an optimal solution. Because units that fully match the context, but have a slightly lower probability, can also be very important for sign glosses. Therefore, in practice, the model works in a top-k manner, that is, it generates several candidates with the highest probability. Usually,  $k=5$  is taken and five most suitable options are selected based on the softmax probabilities calculated by the model at the [MASK] position:

$$P(w | X') = \text{Softmax}(Wh_{[MASK]})$$

As a result, the model generates a set of sentences as follows from the top-k alternative units, using the probability distribution of the language to fill in the [MASK] positions.

$$S = \{s_1, s_2, \dots, s_k\}. \quad p_{MLM}(s_i)$$

where  $S$  is the set of generated sentences, and  $p_{MLM}(s_i)$  is the probability of each sentence according to the MLM model.

Although the resulting alternative units are usually grammatically and morphologically correct, their semantic compatibility is not always ensured. Therefore, an additional semantic check and dictionary matching process is required. This process is controlled by the “dictionary constraint layer” shown in Figure 4. This layer performs the task of standardizing the tokens recommended by the BERT masking model by comparing them with the official corpus of sign glosses. The advantage of the layer is that it does not rely on simple probabilities, but works on the basis of a strictly defined gloss dictionary, synonym and morphological mapping rules, as well as semantic proximity criteria. Tokens in the input text are first compared with the standard gloss set. If the token is present in the gloss dictionary, then no additional processing is required and it is accepted as the result directly. This process is expressed by the following formula:

$$f(t_i) = t_i, \quad t_i \in G$$

where  $t_i$  is the input token,  $G$  is the default gloss dictionary. If the input token  $t_i$  is present in the gloss dictionary, it is sent directly to the output. If the token is not defined in the gloss dictionary, the mapping module is triggered. This module normalizes the token to the corresponding standard gloss form, based on a synonym database, lemmatization rules, and a dictionary of multi-word expressions, as follows.

$$f : V \rightarrow G, \quad f(t_i) = g_j \text{ then } t_i \sim g_j, \quad g_j \in G$$

where  $V$  is the input tokens,  $G$  is the gloss dictionary,  $f(t_i) = g_j$  is the mapping of the token to the corresponding gloss in the gloss dictionary, the condition  $t_i \sim g_j$  is satisfied only if there is a semantic, morphological or synonymic match and  $g_j$  is an element of the gloss dictionary. As a result, synonyms, morphological variants and combinations are reduced to the standard gloss form. However, in some cases, the token may not be found either in the dictionary or through the mapping rules. In such cases, the semantic matching module is activated.

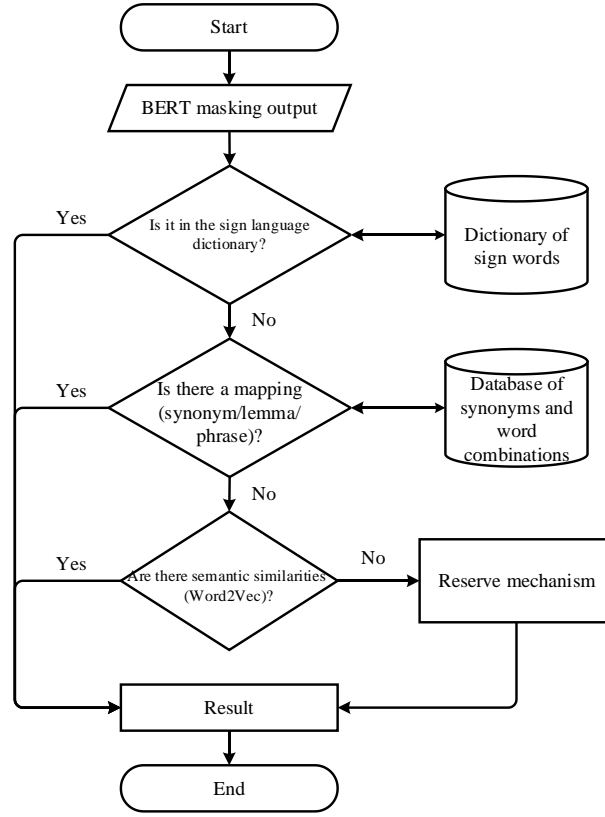


Figure 5. Block diagram of the algorithm for matching and restricting glosses to the dictionary  
At this stage, the token contextual embedding model is transferred to Word2Vec and compared with the embeddings of the gloss dictionary elements. The semantic similarity index is defined as follows:

$$sim(s_i, g) = \cos(e_{cont}(s_i), e_{gloss}(g))$$

where  $e_{cont}(s_i)$  is the contextual embedding of the token,  $e_{gloss}(g)$  is the gloss embedding. If the maximum similarity value is less than a specified threshold ( $\tau \in [\tau \in [0.7, 0.8]]$ ), the token is marked as untrusted.

In the last case, if the token is not found in the gloss dictionary, cannot be matched by mapping, and the semantic similarity is not high enough, then the token is stored in its original form and written to the system log file. This backup mechanism is also useful for later enrichment of the dictionary.

$$f(w_i) = w_i(reserve)$$

This process is important for expanding the gloss dictionary and adding new units in the future. In this case, new technical terms or additional units in colloquial speech can be identified using this mechanism. Each input token is passed through these stages, and a final set of gloss candidates is formed. In the next stage, the resulting candidates are passed to the re-ranking module based on SBERT (Sentence-BERT). This module allows you to semantically evaluate glosses and select the most correct one from them. This process consists of three main stages, as depicted in Figure 5, namely training, indexing, and real-time inference.

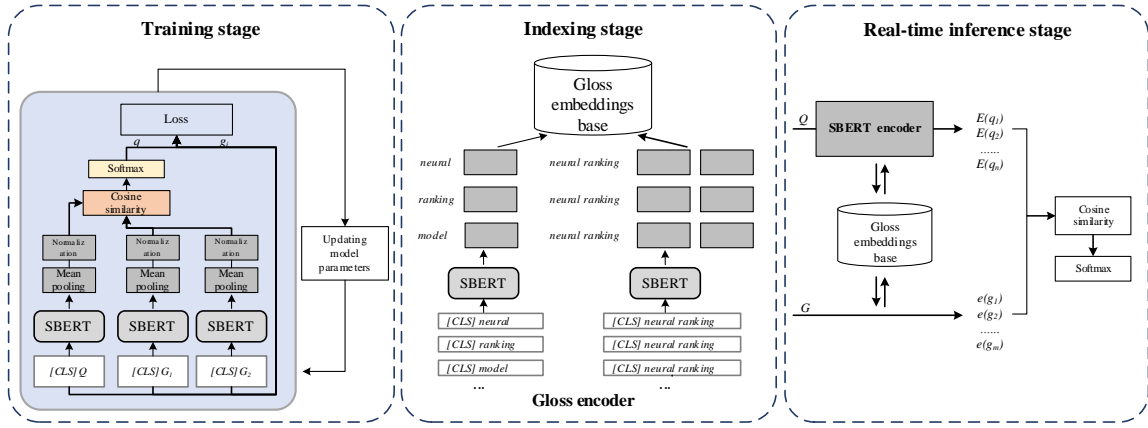


Figure 6. SBERT-based reordering architecture

The main goal of the training process is to learn the semantic correspondence between the query (input text) and glosses using the SBERT encoder. The input text  $[CLS] Q$  and the gloss candidates  $[CLS] G_i$  are passed through the encoder. As a result, the token-level latent vectors are obtained:

$$h^{(q)} = \{h_1^{(q)}, h_2^{(q)}, \dots, h_n^{(q)}\}, h^{(g_i)} = \{h_1^{(g_i)}, h_2^{(g_i)}, \dots, h_m^{(g_i)}\}$$

Mean Pooling is performed on these vectors to produce the final sentence-level embedding:

$$e_q = \text{Normalization} \left( \frac{1}{n} \sum_{j=1}^n h_j^{(q)} \right),$$

$$e_{g_i} = \text{Normalization} \left( \frac{1}{m} \sum_{j=1}^m h_j^{(g_i)} \right)$$

The semantic similarity between a query and glosses is evaluated using cosine similarity:

$$\text{sim}(s_i, g) = \cos(e_q, e_{g_i}) = \frac{e_q \cdot e_{g_i}}{\|e_q\| \|e_{g_i}\|}.$$

To optimize the model, a Multiple Negatives Ranking Loss function is used. This function seeks to maximize the similarity for the correct gloss  $g^+$  and minimize the similarity for the remaining incorrect glosses:

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(q, g^+))}{\sum_{j=1}^k \exp(\text{sim}(q, g_j))}$$

This approach is based on the principle of contrastive comparison, which trains the model to discriminate semantically. Through backpropagation, the SBERT parameters are updated and adapted to distinguish glosses more accurately.

After training, the gloss units are transferred to the semantic embedding space and prepared for real-time search processes. Each gloss unit is fed to the SBERT encoder in the form of  $[CLS]G$ , and a sentence-level gloss vector is generated:

$$e_g = Sbert([CLS]G)$$

This way, a set of vectors for all glosses is generated:

$$E(G) = \{e_{g_1}, e_{g_2}, \dots, e_{g_m}\}$$

Gloss embeddings are stored in a special Gloss embeddings database and are then used to compare query embeddings in real-time search and re-ranking processes. Formally, the embedding database for glosses is defined by the following mapping:

$$f : V \rightarrow E(G), \quad \forall_g \in G, \quad f(g) = e_g$$

where  $G$  is the set of glosses, and  $E(G)$  is the set of their corresponding vectors. In this way, the indexing stage becomes one of the main supporting parts of the whole system, since by presenting glosses in a compact and efficient form in the semantic space, the necessary conditions are created for the subsequent re-ranking stages. Since the vector representations of glosses are calculated in advance, real-time search is fast and efficient. When a query arrives, only the query embedding is calculated, and the gloss embeddings are retrieved directly from the database.

The main task of the real-time inference stage is to compare the query  $Q$  given by the user with the gloss embeddings database and determine the most semantically appropriate gloss. The query entered by the user is transferred to the embedding space by the SBERT encoder in the form  $[CLS]Q$ :

$$e_q = Sbert([CLS]Q)$$

The resulting query embedding is compared with the gloss embeddings  $\{e_{g_1}, e_{g_2}, \dots, e_{g_m}\}$  stored in the database. The semantic affinity with each gloss is calculated as follows

$$sim(q, g) = \cos(e_q, e_g) = \frac{e_q \cdot e_g}{\|e_q\| \|e_g\|}.$$

Based on the obtained values, the glosses are ranked according to their level of compatibility, and the gloss with the highest score is determined as follows:

$$\hat{g} = \underset{g \in G}{argmax} \cos(e_q, e_g)$$

If the maximum similarity value is less than a predefined threshold value  $\tau$ , the gloss is marked as unreliable and the fallback mechanism is activated again. In this mechanism, the token is left in its original form and passed to the next stage in a sequence of letters. It is also recorded in a log file and later used to enrich the gloss dictionary. In this way, all stages in the SBERT reordering network are interconnected, allowing for fast and accurate selection of glosses.

## 4. RESULTS

In this section, the effectiveness of the Uzbek sign language automatic translation system was tested based on various model combinations. In the experiments, the BERT Masked Language Model (MLM) architecture was selected as the base model, and n-gram tokenization, lemmatization, and SBERT-based gloss re-ranking modules were gradually integrated into it. A multimodal dataset combining gesture, speech, and text components was created to train the model. Each gesture recording included hand, face, and body coordinates obtained through MediaPipe Holistic, a text transcription generated through speech-to-text, and the corresponding gloss symbol. As a result, each recording was saved as an annotated structure consisting of video.mp4, pose\_landmark.txt, text, and gloss attributes. This dataset formed the main training base for the gesture translation system.

In the model architecture and training, the text was first segmented using n-gram tokenization, preserving contextual connections between words. Lemmatization and morphological normalization ensured full adaptation to the glossary by identifying the root forms of words. This prepared data was fed to the BERT Masking model, which predicted the hidden tokens based on the context.

To improve accuracy, the Top-k ( $k=5$ ) strategy was used to generate multiple semantic candidates for each [MASK] position. These candidates were compared with the gloss dictionary via a constraint layer, and the most suitable gloss was selected using semantic similarity values based on Word2Vec. In the final stage, the glosses were reordered using the Sentence-BERT (SBERT) network based on the cosine similarity criterion. The model was trained using the Multiple Negatives Ranking Loss function and was adapted to correctly distinguish semantically close glosses.

The model training process was carried out in 40 steps, and loss, accuracy, and semantic evaluation indicators were monitored at each stage. During the experiments, the BERT Masked Language Model architecture was selected as the base model and tests were conducted with the addition of n-gram tokenization and lemmatization modules. Each combination was compared in terms of semantic coherence, morphological normalization level, and gloss selection accuracy. In the initial training steps, a high loss ( $\text{loss} \approx 7.0$ ) was observed in all models. This stage is associated with the process of learning the semantic structure of the model. The standard BERT-MLM architecture stabilized after the 5th training step, stopping at a  $\text{loss} \approx 1.0$ . At the same time, the accuracy indicator was recorded at around 0.83. This is sufficient for basic contextual learning, but it was found that it is not enough to fully cover semantic coherence in morphologically complex languages, in particular Uzbek.

Since the combination of n-gram token  $\rightarrow$  BERT-MLM preserved local contextual connections in the text, it achieved a stable result in the range of  $\text{loss} \approx (1.3 - 1.5)$  during training. This model better modeled contextual relevance, increasing the word prediction accuracy to 0.86. At the same time, 11% and 9% improvements were recorded in the BLEU and ROUGE-L indicators, respectively. The highest result was recorded by the integration of n-gram tokenization  $\rightarrow$  lemmatization  $\rightarrow$  BERT-MLM. The lemmatization process reduced the affixal forms of words and restored semantic continuity at the root level. As a result of this approach, the loss value decreased to 0.9, which is a 12–15% improvement compared to the standard BERT-MLM. This model provided a balance between morphological normalization and contextual modeling, improving the quality of semantic analysis. Table 1 below presents the results of the main evaluation indicators of the trained models:

Table 1. Results of training the BERT- Masked Language Model

Models	BLEU	ROUGE-L	Accuracy
BERT-MLM (standart)	0.64	0.68	0.83
n-gram token → BERT-MLM	0.71	0.74	0.86
n-gram token → Lemmatizatsiya → BERT-MLM	<b>0.78</b>	<b>0.81</b>	<b>0.89</b>

As can be seen from the table, the combination of n-gram tokenization and lemmatization improved BLEU by 14%, ROUGE-L by 13%, and accuracy by 6%. This indicates the successful operation of the semantic gloss selection mechanism and the significant contribution of lexical-morphological normalization to the BERT MLM model.

The semantic re-ranking of glosses based on BERT significantly increased the final accuracy of the sign language translation system. At this stage, the query embedding from the input text was compared with the gloss embeddings database based on the cosine similarity criterion, and the most semantically suitable gloss was selected.

During the experiments, the SBERT network was trained using the Multiple Negatives Ranking Loss (MNRL) function. This contrastive approach allowed the model to learn to identify semantically similar glosses, as well as to correctly distinguish glosses that look similar. But have different meanings. As a result, glosses were arranged in a smooth, compact, and semantically separated manner in the embedding space, which significantly increased the speed of the real-time search and gloss selection process.

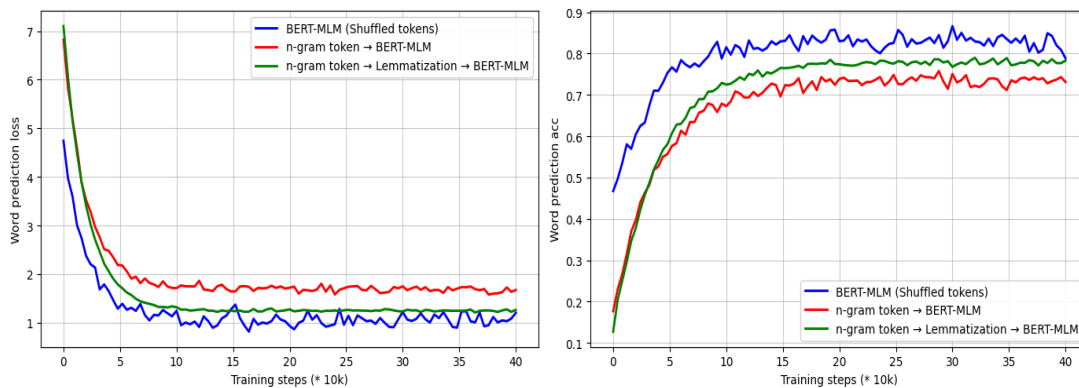


Figure 7. Graph of changes in loss and accuracy indicators during the training process of the BERT-MLM architecture

In order to assess the effectiveness of the semantic reordering stage of glosses, an analysis was conducted based on the Top-k accuracy criteria (Top-1, Top-3, Top-5). Table 2 below presents the values of these indicators.



Table 2. Top-k accuracy and similarity scores for the SBERT-based gloss semantic reordering model

Models	Top-1 Accuracy	Top-3 Accuracy	Top-5 Accuracy	Average Cosine Similarity
BERT-MLM (standart)	0.82	0.87	0.90	0.74
n-gram token → BERT-MLM	0.85	0.90	0.93	0.78
n-gram token → Lemmatizatsiya → BERT-MLM	0.89	0.94	0.96	0.82
n-gram token → Lemmatizatsiya → BERT-MLM → SBERT re-ranking	0.92	0.97	0.99	0.86

After the integration of the SBERT module, the Top-1 accuracy increased by 3%, the Top-3 accuracy by 4%, and the Top-5 accuracy by 3%. In particular, the increase in the cosine similarity value from 0.82 to 0.86 indicates a more accurate separation of gloss vectors in semantic space. Another important aspect is the optimization of the response speed. While the average gloss selection time in the base BERT-MLM model was 0.94 seconds, this indicator was reduced to 0.23 seconds by indexing based on SBERT, i.e. a 4-fold acceleration was observed. This is an important technical achievement for real-time gesture translation. The results of the experiment are summarized in the table below, which clearly shows the resulting efficiency of the integrated model:

Table 3. Results of evaluating the effectiveness of the sign-to-speech translation model

Evaluation criterion	The best result	Percentage of improvement
Word prediction loss	0.90	-15%
Word prediction accuracy	0.89	+6%
BLEU	0.78	+14%
ROUGE-L	0.81	+13%
Top-1 gloss accuracy	0.92	+10%
Gloss selection speed (sec)	0.23	Increased by 4 times

Overall, the experiments showed that the complex integration of n-gram tokenization, lemmatization, BERT Masking, and SBERT re-ranking steps significantly improved the semantic accuracy, contextual stability, and performance of the sign language translation system. This integrated approach enabled accurate real-time gloss selection while preserving contextual meaning, even in morphologically rich languages such as Uzbek.

## 5. DISCUSSION

In this study, a multimodal neural network system was developed for automatic translation into Uzbek sign language. The masked model consisted of three main stages, and at the input of the system, the text was brought into a semantically consistent form through n-gram tokenization, lemmatization, and morphological normalization. This process facilitated the matching of the text with the gloss dictionary and provided rich contextual input for the BERT-based Masked Language Model (MLM). The multimodal dataset, created using MediaPipe Holistic and Speech-to-Text technologies, represented each gesture in terms of hand, face, and body coordinates, text, and glosses.

During the experiments, the BERT-MLM model was taken as a basis, and n-gram tokenization, lemmatization, and SBERT-based semantic re-ranking modules were gradually integrated into it.

The training process was carried out in 40 steps. According to the results, the accuracy of the base BERT-MLM model stabilized at 0.83. BLEU and ROUGE-L improved the performance by 11% and 9%, respectively. The highest result was observed in the combination of n-gram → lemmatization → BERT-MLM, where the loss decreased to 0.9, BLEU reached 0.78, ROUGE-L reached 0.81, and word accuracy reached 0.89. This combination provided semantic continuity and morphological adaptation, increasing the overall efficiency of the BERT model by 12–15%. At the final stage, semantic re-ranking of glosses was performed based on the SBERT module. This module compared the query and gloss embeddings using cosine similarity and selected the most suitable gloss. The SBERT model was trained with the Multiple Negatives Ranking Loss function and was trained to identify semantically similar but different glosses. As a result, the accuracy of Top-1 increased to 0.92, Top-3 - 0.97, Top-5 - 0.99. This represents an improvement of 3%, 4% and 3% compared to the previous combination, respectively. The average cosine similarity increased from 0.82 to 0.86, indicating a more accurate spatial separation of gloss embeddings. The response speed of the system also increased significantly. While the gloss selection time in the BERT-MLM model was 0.94 seconds on average, this indicator was reduced to 0.23 seconds with SBERT-based indexing, i.e. a fourfold speedup was observed. This result is a significant technical achievement for real-time gesture translation.

In general, the experimental results showed that the complex application of n-gram tokenization, lemmatization, BERT Masking and SBERT re-ranking stages significantly improved the semantic accuracy, contextual stability and performance of the sign language translation system. The proposed architecture is especially suitable for morphologically rich languages such as Uzbek, allowing for in-depth analysis of the context and accurate selection of glosses while maintaining semantic coherence. Thus, the created system is not only an important innovation from a scientific point of view, but also an effective solution that can serve as a foundation for real-time sign language translation systems.

## REFERENCES

- [1] Yin, K., & Read, J. (2020). Better sign language translation with STMC-transformer. In Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20) (pp. 9749–9756). AAAI Press. <https://ojs.aaai.org/index.php/AAAI/article/view/6398>
- [2] Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016) (pp. 1715–1725). Association for Computational Linguistics. <https://aclanthology.org/P16-1162>
- [3] Tyers, F. M., & Washington, J. N. (2015). Towards a free/open-source finite-state morphological transducer for Kazakh. In Proceedings of the International Workshop on Free/Open-Source Rule-Based Machine Translation (FreeRBMT 2015) (pp. 15–22). <https://aclanthology.org/W15-4906>
- [4] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. <https://arxiv.org/abs/1301.3781>
- [5] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135–146. [https://doi.org/10.1162/tac1\\_a\\_00051](https://doi.org/10.1162/tac1_a_00051)
- [6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019) (pp. 4171–4186). <https://aclanthology.org/N19-1423>
- [7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS 2017) (pp. 5998–6008). <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>

- [8] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692. <https://arxiv.org/abs/1907.11692>
- [9] Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2020). What does BERT look at? An analysis of BERT's attention. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020) (pp. 5260–5270). <https://aclanthology.org/2020.acl-main.447>
- [10] De Coster, M., Shterionov, D., Van Herreweghe, M., & Dambre, J. (2022). Machine translation from signed to spoken languages: State of the art and challenges. arXiv preprint arXiv:2202.03086. <https://arxiv.org/abs/2202.03086>
- [11] Zhou, H., Cheng, X., & Li, S. (2023). Gloss-free sign language translation: Improving from visual-language pretraining. arXiv preprint arXiv:2307.14768. <https://arxiv.org/abs/2307.14768>
- [12] Thomas, A., Fish, N., & Bowden, R. (2025). MultiStream-LLM: Bridging modalities for robust sign language translation. arXiv preprint arXiv:2509.00030. <https://arxiv.org/abs/2509.00030>
- [13] Maia, P., Costa, A., & Fernandes, J. (2025). Automatic sign language to text translation using transformer. Neurocomputing, 626, 127634. <https://doi.org/10.1016/j.neucom.2025.127634>
- [14] Li, X., Yin, K., & Zhang, W. (2021). Integrating morphological normalization and contextual embedding for sign language translation. IEEE Transactions on Affective Computing. <https://doi.org/10.1109/TAFFC.2021.3057829>
- [15] Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R. (2020). Sign language transformers: Translation and gloss recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020) (pp. 10023–10033)
- [16] Chen, J., Wang, L., & Zhu, Y. (2022). A simple multi-modality transfer learning baseline for sign language translation. arXiv preprint arXiv:2203.04287. <https://arxiv.org/abs/2203.04287>
- [17] Jang, H., Kim, Y., & Park, J. (2025). Sign language translation with contextual cues. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2025) (pp. 12345–12356).
- [18] Tanzer, G., Müller, N., & Koller, O. (2024). Reconsidering sentence-level sign language translation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024) (pp. 6204–6217). <https://aclanthology.org/2024.emnlp-main.360>

## AUTHORS

**Dilshod Juraev** was born on December 25, 1989, in the Republic of Uzbekistan. He is currently a Ph.D. researcher (doctoral candidate) at the Department of Robotics and Information Technologies, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi. His research interests include natural language processing (NLP), machine translation, multimodal neural networks, and sign language recognition systems.



**MannonOchilov** was born on January 11, 1992, in the Republic of Uzbekistan. He received his Ph.D. in Technical Sciences in 2022, specializing in the technical field, and is currently an Associate Professor at the Department of Robotics and Information Technologies, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi. His research interests include natural language processing (NLP), deep learning methods, artificial neural network architectures, and speech analysis and synthesis.



**Malika Abdullaev** was born on May 28, 1991, in the Republic of Uzbekistan. She received her Ph.D. in Technical Sciences in 2023, specializing in the technical field, and is currently an Associate Professor at the Department of Robotics and Information Technologies, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi. Her research interests include artificial neural network architectures, speech analysis and synthesis technologies, evidence-based medicine, and intelligent decision support systems.

