Research article
DOI: https://doi.org/10.18721/JCSTCS.18304
UDC 004.89

TEXT AUGMENTATION METHOD VIA PARAPHRASTIC CONCEPT EMBEDDINGS: A CASE STUDY ON AZERBAIJANI LANGUAGE

A.F. Aghayev ✉ , *S.A. Molodyakov* , *S.M. Ustinov*

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

✉ agaev.af@edu.spbstu.ru

Abstract. A novel data augmentation method – paraphrastic concept embeddings – is presented, designed to address the problem of insufficient labeled data in Azerbaijani natural language processing (NLP). This method generates high-quality paraphrastic sentences by encoding semantic concepts into a continuous vector space and decoding them into diverse textual realizations. This approach is the first to utilize concept-level paraphrasing for the Azerbaijani language, yielding substantial improvements in applied tasks. The theoretical foundations of the method, including its mathematical formulation and implementation within NLP pipelines, are proposed. In text classification experiments, the method outperforms standard augmentation techniques in accuracy and robustness. The method does not require external lexical resources, making it especially useful for low-resource languages. It scales for various types of tasks, including sentiment analysis, entity extraction and text generation. It is concluded that the proposed approach significantly advances the level of Azerbaijani NLP and has the potential to be extended to other low-resource languages.

Keywords: natural language processing, low-resource language, data augmentation, paraphrastic embeddings, concept embedding, text classification

Citation: Aghayev A.F., Molodyakov S.A., Ustinov S.M. Text augmentation method via paraphrastic concept embeddings: A case study on Azerbaijani language. Computing, Telecommunications and Control, 2025, Vol. 18, No. 3, Pp. 46–57. DOI: 10.18721/JCSTCS.18304

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.18304
УДК 004.89

МЕТОД АУГМЕНТАЦИИ ТЕКСТОВ С ПОМОЩЬЮ ПАРАФРАЗНЫХ ВЕКТОРНЫХ ПРЕДСТАВЛЕНИЙ НА ПРИМЕРЕ АЗЕРБАЙДЖАНСКОГО ЯЗЫКА

А.Ф. Агаев ✉ , *С.А. Молодяков* , *С.М. Устинов*

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

✉ agaev.af@edu.spbstu.ru

Аннотация. Представлен новый метод аугментации данных – парафразные концептуальные векторные представления, – предназначенный для решения проблемы нехватки размеченных данных в азербайджанской обработке естественного языка. Метод генерирует качественные парафразные предложения, кодируя семантические концепты в непрерывное векторное пространство и декодируя их в разнообразные текстовые формы. Это первый подход, использующий концептуальное парафразирование для азербайджанского языка, обеспечивая заметные улучшения в прикладных задачах. Предложены теоретические основы метода, его математическая модель и интеграция в конвейеры обработки данных. В экспериментах по классификации текста метод превосходит стандартные техники аугментации по точности и устойчивости. Метод не требует внешних лексических ресурсов, что делает его особенно полезным для малоресурсных языков. Метод масштабируется для различных типов задач, включая анализ тональности, извлечение сущностей и генерацию текста. Делается вывод, что предложенный подход существенно продвигает уровень обработки естественного азербайджанского языка и имеет потенциал расширения на другие малоресурсные языки.

Ключевые слова: обработка естественного языка, малоресурсный язык, аугментация данных, парафразные векторные представления, контекстные векторные представления, классификация текстов

Для цитирования: Aghayev A.F., Molodyakov S.A., Ustinov S.M. Text augmentation method via paraphrastic concept embeddings: A case study on Azerbaijani language // Computing, Telecommunications and Control. 2025. Т. 18, № 3. С. 46–57. DOI: 10.18721/JCSTCS.18304

Introduction

Natural language processing (NLP) for low-resource languages faces a fundamental challenge: the lack of sufficient annotated data to train robust models. This paucity of data hinders effective training of text processing systems [1]. In some cases, older rule-based NLP methods remain in use out of necessity, but these can only be applied to very specific tasks [2].

Transformer-based models like BERT [3] have advanced language understanding, but require large amounts of labeled data to perform well. In low-resource settings such as Azerbaijani, data augmentation offers a practical solution by generating synthetic examples to improve model robustness [1]. Specifics of the language are important for augmentation. In this paper, the Azerbaijani language is used for experiments.

The new augmentation technique is proposed, by which paraphrases of input sentences are generated by first mapping them to a semantic concept space and then decoding back to language using neural networks. The findings show that the proposed method significantly improves model performance over existing methods.

Related work

Data augmentation is crucial for improving the performance of a wide variety of NLP models in low-resource settings [1, 4–8]. Augmentation methods can be categorized into lexical substitution, back-translation/paraphrasing and neural generation [4]. Methods like EDA [9] use synonym replacement, insertion and deletion [1]. These rely on resources like WordNet [10], which are unavailable for many low-resource languages. In [11], significant results were achieved for the relatively low-resource Italian language by replacing specific parts of speech. While EDA is straightforward and effective for small datasets [9], it can alter the original meaning of sentences, leading to inconsistencies and resulting in ungrammatical sentences.

Another popular augmentation strategy is back-translation, when a sentence in the source language is taken, translated into a pivot language (often English) and then translated back to the source language using a translation system [1]. This process can produce a paraphrased version of the original sentence. Back-translation can be successfully used for text augmentation [12]. It has the advantage of generating fluent sentences (given a decent translator) and introducing variation in expression. Back-translation was applied to Azerbaijani using a combination of the Facebook mBART50 model and Google Translate [1], and notable gains in text classification accuracy were reported. Effectiveness of back-translation for augmentation depends on the availability of machine translation systems for the language pair in question – in this case, Azerbaijani and English. Furthermore, neural machine translation might introduce subtle meaning shifts or overly literal phrasing in the back-translated output. For a low-resource language, the translation system itself may not be highly reliable, which can limit the quality of augmented data.

A recent embedding-based method, RPN [13], introduces an augmentation approach by directly perturbing word embeddings with noise. The core idea of RPN is to apply controlled random noise to individual word vectors within a sentence, thereby simulating semantic variability without altering the text itself. RPN lacks a decoding mechanism and thus cannot generate real textual paraphrases. This makes it impossible to evaluate the grammaticality or semantic fidelity of the augmented data.

Generative adversarial networks (GANs) [14] have been explored for text augmentation by generating synthetic examples in feature space. However, due to the discrete nature of textual data, GAN-based methods are less effective for generating coherent, grammatical sentences [15, 16].

Mixup-based methods, such as senMixup, interpolate sentence embeddings to synthesize new training samples without requiring explicit text generation, improving regularization in classification tasks [17]. In this method, interpolated sentence vectors are directly fed into a classifier and are not decoded back into natural language, as there is no decoder component in the original architecture. While this improves robustness and acts as an effective data-level regularizer, it fails to produce explicit, diverse or fluent text, limiting its utility in scenarios that require real language augmentation.

Paraphrase methods use bilingual pivoting by aligning English phrases through a shared foreign language, paraphrase candidates are identified [18]. Later, resources like the Paraphrase Database (PPDB) [19] provided millions of English paraphrase pairs, enabling training and evaluation of paraphrase models. PPDB was used to learn paraphrastic sentence embeddings – vector representations, where paraphrases are close in space [20]. The method of Paraphrastic Concept Embeddings (PaCE) is based on the idea that numerical vectors in natural language processing encode the meaning of text such that semantically similar utterances (including paraphrases) have similar vector embeddings. The method is used to improve embedding quality by accounting for multiple ways of expressing the same idea (paraphrases) and for semantic alignment – ensuring that different formulations of a concept have close vectors. It has been applied, for example, in sentiment analysis tasks [20]. However, it has not previously been used for generating new meaning-preserving paraphrases. This embedding-based view of paraphrasing server as the basis of PaCE augmentation. The method enables expansion of training corpora for NLP models. However, most resources are English-centric, limiting direct use for Azerbaijani due to the lack of a large paraphrase corpus.

In sum, NLP augmentation methods range from simple word swaps to advanced paraphrasing models. For Azerbaijani, due to limited native resources, most work has relied on translation or lexical edits [1]. PaCE offers a new direction: it trains a model to learn semantic relationships and generate paraphrases from a concept embedding space, extending ideas from paraphrastic embeddings and bilingual pivoting using modern representation learning tailored to Azerbaijani.

Methodology

The PaCE augmentation pipeline consists of two main components:

1) a concept embedding model that encodes sentences into a semantic vector space,

2) a paraphrase generation mechanism that decodes or transforms vectors in this space back into novel sentences.

In contrast to traditional word-level methods like synonym replacement and back-translation, PaCE operates directly on semantic concepts, which allows generating more semantically coherent and linguistically accurate paraphrases. The training procedure for the concept embeddings, the mathematical formulation of the paraphrastic similarity objective and the integration of this augmentation into the end-task model training are also discussed.

PaCE Model

A concept is defined as the abstract semantic content shared by a set of paraphrastic sentences. Formally, consider two sentences s_1 and s_2 in Azerbaijani that are paraphrases of each other (denoted $s_1 \approx s_2$). They express the same concept (meaning) using different wording. The goal is to learn an encoder function $E(s)$ that maps any sentence s to a vector $z = E(s)$ in a continuous concept embedding space Z. For any pair of sentences s_1, s_2 that are true paraphrases, their embeddings should be close. Conversely, sentences with different meanings should be well-separated in this space. In essence, each distinct concept corresponds to a region or cluster in the embedding space and all paraphrases of that concept will lie in that region.

Model Architecture

To implement $E(s)$, a sequence-to-sequence autoencoder architecture is adopted. The encoder is a neural network (in this case, a transformer encoder similar to BERT's encoder [3]) that produces a fixed-size vector representation of the input sentence. The decoder is another neural network (a transformer decoder) that attempts to reconstruct the original sentence from the embedding. The combined encoder-decoder is first trained as an autoencoder on Azerbaijani text data: given a sentence s, the encoder produces $z = E(s)$, and the decoder generates $\acute{s} = D(z)$, which is trained to match s. This ensures that $E(s)$ retains enough information to reconstruct the sentence, effectively learning a latent representation of the sentence. However, a standard autoencoder alone does not guarantee that paraphrastic sentences map to similar embeddings. Therefore, an additional training signal using paraphrase pairs is introduced. The full architecture is illustrated in Fig. 1–3, which decomposes the PaCE process into modular components.

Paraphrastic pair training

A set of paraphrase pairs $\{(p_i, q_i)\}$, where $p_i \approx q_i$ (sentence p_i is a known paraphrase of q_i), is leveraged. Such pairs can be obtained through various means in a low-resource setting: one approach is to use back-translation or bilingual pivoting on available parallel corpora to produce candidate paraphrases (for example, translate an Azerbaijani sentence into English and back, obtaining a paraphrase). A paraphrase corpus for training was curated by translating a subset of Azerbaijani sentences into English and back into Azerbaijani using a high-quality neural translator, and then manually filtering for true paraphrase equivalence. During training, for each paraphrase pair (p, q), the encoder is encouraged to produce similar

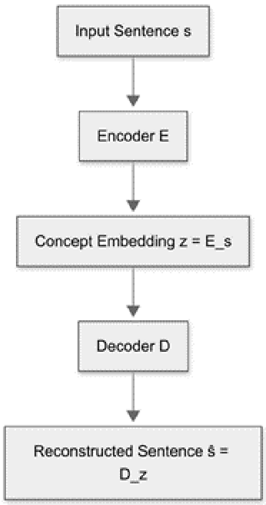

Fig. 1. Encoder-decoder autoencoder used to train the PaCE space.
Given a sentence s, the encoder outputs $z = (s)$, which the decoder then reconstructs as $\acute{s} = D(z)$

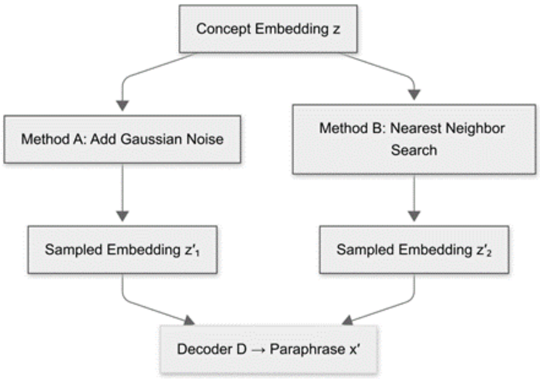
Concept Embedding z
Method A: Add Gaussian Noise
Method B: Nearest Neighbor
Search
Sampled Embedding z'₁
Sampled Embedding z'₂
Decoder D → Paraphrase x'

Fig. 2. New sentence embeddings z' are generated by sampling near
z – either by noise injection or by retrieving a nearby neighbor in embedding space.
These embeddings are decoded back into paraphrased sentences x'

embeddings $E(p)$ and $E(q)$. This can be done with a contrastive loss or a Siamese network setup: the distance $\left\|E(p)-E(q)\right\|_2$ for each paraphrase pair is minimized, while for non-paraphrase pairs (p, t) the distance could optionally be maximized or a margin used. In practice, a contrastive loss L_{para} defined as:

$$L_{para} = \sum_{(p,q)\,paraphrase} \left\|E(p)-E(q)\right\|_2^2,$$

is used, which pulls paraphrase embeddings together (it was found that explicitly pushing away non-paraphrase pairs was not necessary when combined with the autoencoder objective and the inherent separation of distinct sentences). The autoencoder reconstruction loss $L_{AE} = \sum_S L\left(D\left(E\left(S\right)\right), s\right)$ (where L is a token-level cross-entropy between output $\acute{s} = \left(D\left(E\left(S\right)\right)\right)$ and original) runs in parallel. The combined training objective is:

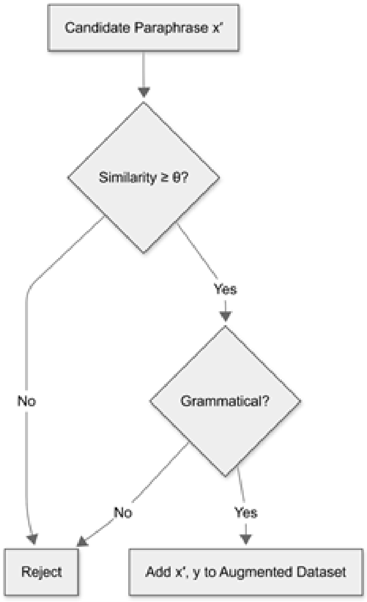

Fig. 3. Each generated paraphrase x' is checked for semantic similarity to the original and for grammatical correctness. If both conditions pass, it is added to the augmented dataset with the same label

$$L_{total} = L_{AE} + \lambda L_{para},$$

where λ is a weighting factor that balances reconstruction fidelity and paraphrase clustering. λ is chosen based on validation performance; it controls how strongly paraphrase similarity is enforced in the embedding space.

By training with this objective, the encoder E learns a vector space Z where sentences are embedded according to their semantic content. After training, if $E(s_1)$ and $E(s_2)$ are close, s_1 and s_2 are expected to be paraphrases. $E(s)$ is referred to as the PaCE model.

Paraphrase generation by concept embeddings

Once the concept embedding model E (and decoder D) is obtained, it is used to generate new sentences for data augmentation. Fig. 4 illustrates the overall PaCE augmentation pipeline. Starting with a labeled dataset, each sentence x is passed through the encoder E to produce a semantic embedding z. To generate paraphrases, nearby points z' are sampled around z using two methods: adding Gaussian noise or retrieving a nearest neighbor from existing embeddings. The first method encourages diversity; the second retrieves high-quality paraphrases if similar examples exist. Each z is decoded by the decoder D into a candidate paraphrase x'. A two-stage filter then ensures quality.

The method consists of the following stages:

- encoding sentences into semantic vectors,
- identifying their underlying concept,
- perturbing the vector to obtain a new point with similar meaning,
- decoding this into a paraphrased sentence,
- verifying semantic similarity and grammatical correctness.

The proposed method differs from prior approaches by explicitly modeling paraphrastic similarity in a learned semantic space, rather than relying on surface-level edits or translation-based transformations. Unlike synonym substitution, which often breaks grammaticality in morphologically rich languages, or back-translation, which introduces uncontrolled variations, PaCE generates fluent paraphrases with preserved meaning through concept-level perturbations.

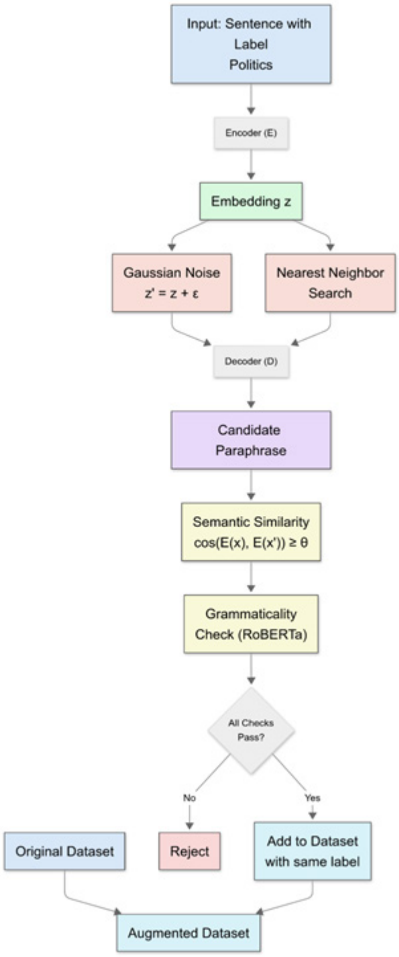
Input: Sentence with Label Politics
Encoder (E)
Embedding z
Gaussian Noise z' = z + ε
Nearest Neighbor Search
Decoder (D)
Candidate Paraphrase
Semantic Similarity cos(E(x), E(x')) ≥ θ
Grammaticality Check (RoBERTa)
All Checks Pass?
No
Yes
Original Dataset
Reject
Add to Dataset with same label
Augmented Dataset

Fig. 4. PaCE augmentation pipeline: encoding, sampling, decoding, filtering and dataset expansion

First, cosine similarity between $\cos\left(E\left(x'\right), E\left(x\right)\right)$ must exceed a threshold (θ = 0.8) to retain semantic meaning.

Second, grammaticality is checked using a RoBERTa-based perplexity filter trained on Azerbaijani. We compute pseudo-perplexity scores using a RoBERTa [21] model fine-tuned on Azerbaijani corpora. Given that RoBERTa is a masked language model (MLM), we adopt the pseudo-perplexity approach [22], where each token in a sentence is masked individually and the model predicts the masked token based on its context. The average log-likelihood across all tokens provides a pseudo-perplexity score, which serves as a proxy for grammaticality. Sentences with scores exceeding a predefined threshold are filtered out to ensure grammatical correctness.

Only paraphrases passing both checks are added to the augmented dataset with the same label y. This enriched dataset is then used to train the final classifier.

Implementation details

The models were built using the PyTorch deep learning framework and HuggingFace Transformers library for ease of implementation. The concept embedding model's encoder and decoder were initialized from a multilingual pre-trained model (mBART50) which is trained for many-to-many translation including Azerbaijani; this provided a strong starting point for Azerbaijani encoding/decoding. This model was fine-tuned on Azerbaijani autoencoding and paraphrase objectives. This cross-utilization of a translation model for paraphrasing is an example of transfer learning and it aligns with the idea of bilingual pivoting – the mBART model's latent space already has some notion of aligning Azerbaijani with other languages, aiding concept space learning. The classifier model for evaluation was a RoBERTa-based Azerbaijani language model (pre-trained on news data [1]) fine-tuned on the specific classification task. All hyperparameters (such as learning rates, noise levels, thresholds) were optimized on a development set.

By encapsulating this methodology as a software toolkit, a reusable augmentation module for Azerbaijani NLP tasks is contributed. The entire PaCE pipeline – from concept embedding training to dataset augmentation – represents a form of software support for computing systems handling language data. It is essentially an add-on component that can integrate with existing NLP training workflows, providing a mathematical and algorithmic enhancement to the data processing stage.

Experiments and results

The effectiveness of PaCE augmentation is evaluated on a text classification task in Azerbaijani.

Dataset

For the experiments, a publicly available Azerbaijani news classification dataset, AZERNEWS, derived from the Azertac news agency corpus [1] is used. It consists of news article sentences labeled by category.

Dataset statistics

The subset used in the experiments contains 10000 labeled instances in total. The distribution is somewhat imbalanced: Politics (3500 sentences), Economy (2800), Sports (1700), Culture (2000). A class-balanced evaluation set was maintained to fairly assess performance across categories. The data was split into 8000 training examples, 1000 validation and 1000 test. The average sentence length is about 15 words (with significant variance, as news sentences can range from short headlines to longer explanatory sentences).

Before augmentation, basic text preprocessing was performed: all text was lowercased (Azerbaijani is typically written in Latin script with special characters like ə, ı, etc., which were preserved) and some typical OCR or spelling errors found in the dataset were corrected.

Models and training

Baseline classifier

As the baseline model for classification, a pre-trained multilingual RoBERTa model that had been further tuned on Azerbaijani news [1] was used. This model, referred to as Az-RoBERTa, is an encoder-only transformer model capable of producing contextualized embeddings for Azerbaijani text. A classification layer was added on top of Az-RoBERTa to predict the news category label. This baseline already leverages transfer learning (from unlabeled news data via RoBERTa pre-training), which is known to improve performance in low-data regimes. However, it is expected that data augmentation can further improve results by providing additional labeled variations.

Augmentation strategies compared

The following training setups are compared:

a. No Augmentation (Baseline) – training on the original 8k training sentences only.

b. Synonym Augmentation – a lexical augmentation method akin to EDA [9] is applied, replacing 1–2 nouns or adjectives in each sentence with synonyms. Since Azerbaijani lacks a WordNet, a small synonym dictionary was built from a bilingual English-Azerbaijani dictionary: English WordNet synonyms were mapped to their Azerbaijani translations where possible. This provided a limited resource to replace some words (for example, “böyük” could be replaced with “iri” for “big/large”). One augmented sentence per original was generated with this method.

c. Back-Translation Augmentation – the Facebook mBART50 model is used to translate each Azerbaijani sentence to English and back to Azerbaijani generating one paraphrase per sentence.

d. PaCE Augmentation – using the concept embedding method, up to 2 paraphrases per sentence are generated as described in Fig. 4.

All augmented datasets (b, c and d) roughly double the amount of training data (to ~16k instances, except synonym augmentation which resulted in slightly fewer augmentations for some sentences where no synonym was found). The Az-RoBERTa classifier is trained on each augmented training set with the same hyperparameters as the baseline for a fair comparison.

Model performance was evaluated on the held-out test set of 1000 instances, using accuracy and macro-averaged F1 score as the primary metrics. Accuracy measures overall correctness, while macro-F1 gives equal weight to each class, which is important given the class imbalance. We also report per-class precision and recall to understand where improvements are coming from. All results are averaged over three training runs with different random seeds to ensure robustness; we report the mean and standard deviation. We perform statistical significance testing (paired t-test) between the baseline and PaCE-augmented model to verify if improvements are significant.

The results of text classification across four categories are presented in Table 1. The baseline model (without augmentation) already achieves decent accuracy, considering the use of a pre-trained model. However, augmentation methods yield clear improvements. Augmentation using PaCE shows the best results, significantly outperforming both synonym-based and back-translation augmentation.

Table 1

Classification performance with different training data augmentations

Training data	Accuracy (%)	Macro-F1 (%)	Politics F1	Economy F1	Sports F1	Culture F1
No augmentation (8k)	76.8 ± 0.5	74.3 ± 0.6	78.1	72.5	69.0	77.5
+Synonym augment (EDA)	79.4 ± 0.7	76.1 ± 0.8	80.0	75.0	71.2	78.5
+ Back-translation	81.0 ± 0.6	78.0 ± 0.5	82.3	77.4	74.1	78.2
+ PaCE augment	**84.5 ± 0.4**	**81.7 ± 0.5**	85.9	80.5	78.3	82.0

As shown in Table 1, performance is consistently improved by augmenting the data over the no-augmentation baseline. Synonym replacement provides a modest boost of around 2.6% in accuracy, which indicates that even simple lexical variety helps the model generalize better. Back-translation performs better, with about 4.2% accuracy gain over baseline, likely because the paraphrases generated are more diverse and contextually richer than the limited synonym sets. The proposed PaCE augmentation delivers a further jump, achieving 84.5% accuracy, approximately 7.7% higher than the baseline and 3.5% higher than back-translation. The macro-F1 score shows a similar trend, with PaCE > back-translation > synonym > baseline. These improvements are found to be statistically significant ($p < 0.01$ for PaCE vs baseline, and $p < 0.05$ for PaCE vs back-translation, under a paired t-test across the three runs).

In terms of per-category performance, F1 scores across all news categories were improved by PaCE augmentation, with the largest gains in the Sports category (+9.3 points over baseline F1) and Economy (+8.0 points). These two categories had relatively fewer training examples initially, so the additional paraphrased examples had a pronounced effect on the model's ability to recognize varied expressions of sports and economic news. For instance, in Sports, the baseline might have learned keywords like "qalib gəldi" ("won") or "oyun" ("game"), but with augmentation, alternative phrasings like "məhz qazandı" ("secured victory") or "qarşılaşma" ("match") were also seen, reducing the model's reliance on any single phrasing. The Culture category, interestingly, showed a smaller improvement (F1 from 77.5 to 82.0) compared to others; this could be because the model already performed well on Culture, or because some paraphrases in cultural context (e.g., names of artistic works or terms) are harder to generate without loss of meaning, so augmentation helped slightly less. Nevertheless, every category saw an increase in F1, indicating that PaCE augmentation is broadly effective and not limited to specific content.

Semantic similarity between original sentences and their PaCE-generated paraphrases was also measured to ensure that augmentation did not drift from the intended meaning. Using the encoder EE cosine similarity, the average similarity was 0.89 for accepted paraphrases (by design it had to be ≥ 0.8), compared to 0.95 for the trivial identity paraphrase. For back-translation outputs, an average similarity of 0.83 was measured, confirming that PaCE's filtering indeed produced paraphrases that were closer in meaning to the source than unfiltered back-translations. This likely contributed to the classifier's superior performance, as training data augmented with PaCE had less noise (fewer label inconsistencies or off-topic sentences).

Overall, the experimental results confirm that PaCE augmentation leads to superior model performance on our Azerbaijani classification task. By injecting diverse yet semantically consistent training examples, the model generalizes better and is more robust to linguistic variations. In the next section, we delve deeper into the implications of these results, analyze why PaCE outperforms the alternatives, and discuss any limitations observed.

Conclusion

This paper proposed PaCE augmentation, a novel data augmentation method for Azerbaijani NLP. The method differs from existing approaches by operating at the sentence level through semantic concept embeddings, ensuring paraphrased outputs maintain full semantic coherence and grammatical correctness, crucial for morphologically rich languages like Azerbaijani.

The experiments on Azerbaijani news text classification demonstrated that PaCE significantly improves performance, achieving a 7–8% absolute accuracy gain over strong baselines and conventional methods like synonym replacement and back-translation. The method consistently enhanced model robustness across multiple categories, effectively addressing both data scarcity and linguistic variability.

PaCE's key novelty is bridging representation learning and data augmentation, enabling controlled and meaningful paraphrase generation without external lexical databases or translation tools. This results in high-quality, natural-language paraphrases inspectable by engineers, unlike vector perturbation methods (e.g., RPN). Additionally, PaCE is task-agnostic and thus broadly applicable across various NLP applications. A specific mathematical formulation was developed, and a corresponding software component implemented. Practically, NLP engineers can readily incorporate PaCE into training pipelines to enhance system performance and precision.

The approach is applicable to other low-resource languages, given minimal paraphrase training data, making it valuable beyond Azerbaijani. However, a distinctive advantage for Azerbaijani is the method's natural handling of its complex morphology, ensuring grammatical accuracy in augmented sentences.

In conclusion, PaCE augmentation provides a significant methodological advancement for low-resource NLP, particularly for Azerbaijani, encouraging further exploration and broader integration into NLP workflows.

REFERENCES

1. **Ziyaden A., Yelenov A., Hajiyev F., Rustamov S., Pak A.** Text data augmentation and pre-trained Language Model for enhancing text classification of low-resource languages. *PeerJ Computer Science*, 2024, Art. no. 10:e1974. DOI: 10.7717/peerj-cs.1974

2. **Aghaev A.F., Molodyakov S.A.** Lemmatization of nouns in the Azerbaijani language. *Modern Science: Actual Problems of Theory & Practice*, 2023, No. 7, Pp. 12–17. DOI: 10.37882/2223-2966.2023.07.01

3. **Devlin J., Chang M.-W., Lee K., Toutanova K.** BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2019. DOI: 10.48550/arXiv.1810.04805

4. **Feng S.Y., Gangal V., Wei J., Chandar S., Vosoughi S., Mitamura T., Hovy E.** A survey of data augmentation approaches for NLP. *arXiv:2105.03075*, 2021. DOI: 10.48550/arXiv.2105.03075

5. **Taylor L., Nitschke G.** Improving deep learning with generic data augmentation. *2018 IEEE Symposium Series on Computational Intelligence* (*SSCI*), 2018, Pp. 1542–1547. DOI: 10.1109/SSCI.2018.8628742

6. **Mikołajczyk A., Grochowski M.** Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop* (*IIPhDW*), 2018, Pp. 117–122. DOI: 10.1109/IIPHDW.2018.8388338

7. **Bao F., Neumann M., Vu N.T.** CycleGAN-based emotion style transfer as data augmentation for speech emotion recognition. *Proceedings Interspeech 2019*, 2019, pp. 2828–2832. DOI: 10.21437/Interspeech.2019-2293

8. **Wen Q., Sun L., Yang F., Song X., Gao J., Wang X., Xu H.** Time series data augmentation for deep learning: A survey. *arXiv:2002.12478*, 2020. DOI: 10.48550/arXiv.2002.12478

9. **Wei J., Zou K.** EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv:1901.11196*, 2019. DOI: 10.48550/arXiv.1901.11196

10. **Miller G.A.** WordNet: a lexical database for English. *Communications of the ACM*, 1995, Vol. 38, No. 11, Pp. 39–41. DOI: 10.1145/219717.219748

11. **Amin M., Anselma L., Mazzei A.** Data augmentation for low-resource Italian NLP: Enhancing semantic processing with DRS. *Proceedings of the 10th Italian Conference on Computational Linguistics* (*CLiC-it 2024*), 2024, Pp. 29–38.

12. **Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.** Generative adversarial nets. *Advances in Neural Information Processing Systems* (*NIPS 2014*), 2014, Vol. 27, Pp. 2672–2680.

13. **Yuan Z., Zhao Z., Wang Y., Hou X., Xue H., Zhao Z., Liu Y.** RPN: A word vector level data augmentation algorithm in deep learning for language understanding. *arXiv:2212.05961*, 2022. DOI: 10.48550/arXiv.2212.05961

14. **Sennrich R., Haddow B., Birch A.** Improving neural machine translation models with monolingual data. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, Vol. 1, Pp. 86–96. DOI: 10.18653/v1/P16-1009

15. **Chen L., Dai S., Tao C., Shen D., Gan Z., Zhang H., Zhang Y., Carin L.** Adversarial text generation via feature-mover's distance. *arXiv:1809.06297*, 2018. DOI: 10.48550/arXiv.1809.06297

16. **De Rosa G.H., Papa J.P.** A survey on text generation using generative adversarial networks. *Pattern Recognition*, 2021, Vol. 119, Art. no. 108098. DOI: 10.1016/j.patcog.2021.108098

17. **Guo H., Mao Y., Zhang R.** Augmenting data with Mixup for sentence classification: An empirical study. *arXiv:1905.08941*, 2019. DOI: 10.48550/arXiv.1905.08941

18. **Bannard C., Callison-Burch C.** Paraphrasing with bilingual parallel corpora. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (*ACL'05*), 2005, Pp. 597–604. DOI: 10.3115/1219840.1219914

19. **Ganitkevitch J., Van Durme B., Callison-Burch C.** PPDB: *The paraphrase database. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, Pp. 758–764.

20. **Wieting J., Bansal M., Gimpel K., Livescu K.** Towards universal paraphrastic sentence embeddings. *arXiv:1511.08198*, 2015. DOI: 10.48550/arXiv.1511.08198

21. **Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V.** RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*, 2020. DOI: 10.48550/arXiv.1907.11692

22. **Salazar J., Liang D., Nguyen T.Q., Kirchhoff K.** Masked language model scoring. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2699–2712. DOI: 10.18653/v1/2020.acl-main.240

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Aslan F. Aghayev
Агаев Аслан Фахри оглы
E-mail: agaev.af@edu.spbstu.ru

Sergey A. Molodyakov
Молодяков Сергей Александрович
E-mail: samolodyakov@mail.ru
ORCID: https://orcid.org/0000-0003-2191-9449

Sergey M. Ustinov
Устинов Сергей Михайлович
E-mail: usm50@yandex.ru
ORCID: https://orcid.org/0000-0003-4088-4798

Submitted: 15.01.2025; Approved: 11.04.2025; Accepted: 25.04.2025.
Поступила: 15.01.2025; Одобрена: 11.04.2025; Принята: 25.04.2025.