Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 5, October 2025, pp. 3778∼3787
ISSN: 2302-9285, DOI: 10.11591/eei.v14i5.9960

Fast lightweight convolutional neural network for Turkish sentiment analysis

Saed Alqaraleh[1], Abdul Hafiz AbdulHafiz[2]
[1]Department of Data Science and Artificial Intelligence, College of Information Technology, Mutah University, Karak, Jordan
[2]Department of Computer Science, College of Computer Science and Information Technology, King Faisal University, Al-Ahsa, Kingdom of Saudi Arabia

Article Info

Article history:

Received Jan 26, 2025
Revised Aug 12, 2025
Accepted Sep 1, 2025

Keywords:

Natural language processing
Neural networks
Sentiment analysis
Text mining
Text processing

ABSTRACT

This study presents a fast, lightweight, and high-performing fast convolutional neural network (Fast CNN) model tailored for Turkish sentiment analysis (SA). The agglutinative morphology of Turkish, combined with the limited availability of high-quality linguistic resources, introduces significant challenges for conventional approaches. To address these issues, we propose a streamlined Fast CNN architecture consisting of an embedding layer, global max-pooling, dropout, and fully connected layers. Despite its simplicity, the model outperforms seven state-of-the-art convolutional neural network (CNN)-based systems across four benchmark Turkish sentiment datasets. It achieves an average area under the curve (AUC) of 0.94, representing a 6.8% improvement over the strongest baseline and a gain of over 80% relative to several deeper architectures. In addition to its superior accuracy, the model demonstrates reduced computational complexity, making it well-suited for real-world deployment in resource-constrained environments. Potential applications include customer feedback mining and digital marketing analytics in Turkish-language domains.

Corresponding Author:

Saed Alqaraleh
Department of Data Science and Artificial Intelligence, College of Information Technology, Mutah University
Karak, Jordan
Email: saed.alqaraleh@mutah.edu.jo

1. INTRODUCTION

Feelings, opinions, and sentiments influence most human activities and play a critical role in shaping individual decisions. Sentiment analysis (SA)—also referred to as opinion mining (OM) or feelings mining (FM)—is a subfield of natural language processing (NLP) concerned with identifying and extracting subjective information from textual data, including opinions about products, services, and individuals. The widespread use of social media platforms (e.g., Facebook, Instagram, and Twitter) has made user-generated content a rich resource for SA. Businesses increasingly analyze this data to understand consumer perceptions and adapt their strategies accordingly. In this context, SA techniques have become essential tools in domains such as marketing, customer feedback, and political communication. SA methods are typically categorized as lexicon-based or machine–learning–based. Lexicon-based methods rely on predefined polarity lexicons to compute semantic orientation scores. For Turkish, a commonly used resource is SentiTurkNet [1], which has been used in lexicon-based and hybrid SA pipelines. For other agglutinative languages, such as Telugu, adaptations based on SentiWordNet have also been employed [2].

In contrast, machine learning-based methods—particularly deep learning architectures, such as convolutional neural networks (CNNs)—have demonstrated superior performance in recent years [3]-[11]. However, languages such as Turkish and Korean pose significant challenges due to their agglutinative morphology and limited annotated resources. As a workaround, some researchers have translated English lexicons into Turkish [12], although this compromises linguistic fidelity. Recent works continue to explore SA in morphologically complex and under-resourced languages, including Korean, Telugu, and Turkish, using both neural and lexicon-based approaches [2], [13]. Despite progress in English-language SA systems, relatively few CNN-based models have been proposed for Turkish [14]-[17]. Notably, only two of these offer publicly available architectures suitable for reimplementation [15], [16]. This paper presents a lightweight yet high-performing fast convolutional neural network (Fast CNN) model tailored to the Turkish language. The architecture includes an embedding layer, a global max-pooling layer, dropout, and two fully connected layers. Despite its simplicity, the model outperforms seven state-of-the-art CNN models across four benchmark Turkish SA datasets. As demonstrated in our experiments, the model achieves higher accuracy while reducing computational overhead, making it well-suited for deployment in real-world, resource-constrained environments. Table 1 shows validation examples from four datasets that our Fast CNN correctly classifies while all seven baseline CNNs fail. These idiomatic, ironic/sarcastic, and sentimentally nuanced cases indicate stronger robustness to real-world Turkish usage.

Table 1. Validation texts (four datasets) correctly classified by our Fast CNN but misclassified by all seven baseline CNNs, including idiomatic, ironic/sarcastic, and sentimentally nuanced Turkish

Dataset	Validation sample (TR)	Meaning (EN)	Class
HUMIR	*Büyük hayal kırıklığına uğradım. Son daha baştan anlaşılıyor. Kötü işlenmiş zayıf espriler filan filan. Bence kesinlikle para harcanmamalı.*	I was highly disappointed. The movie's end was evident from the very start. It has poorly crafted jokes. It's not worth the money.	Negative
TWT	*Tarımda GAP neyse spordada Ş.Urfa hipodrumunun aydınlatılması devrimdir ne yazikki ne basında nede TV de ilgi gördü :(*	What GAP is in architecture, the floodlights in Urfa city's hippodrome are revolutionary in sports. Unfortunately, this didn't attract much media attention.	Negative
Movie reviews	*Süper bir film!Medya bu kadar güzel mi taşlanır!Harika!J.Carrey'in de performansı büyüleyici doğrusu!*	It's an excellent movie. Fantastic! Such a beautiful allusion to the media! And to be honest, J.Carrey displayed a mesmerizing performance.	Positive
17BinTweet	*İyi ki varsınız iyi ki bizimlesiniz diyen turkcell müşteri hizmetleri de olmasa yalnızım dicektim.*	If it weren't for Turkcell customer service telling me they are glad and appreciate having me by their side, I probably would feel lonely.	Positive

2. REVIEW OF THE RECENT DEVELOPMENTS IN OPINION AND FEELINGS MINING

This section reviews recent developments in OM methodologies, with a focus on the Turkish language context. Early studies employed support vector machines (SVM) with n-gram features for sentiment classification tasks [10], also incorporating part-of-speech tagging, spell-checking, and stemming. Subsequent work applied naive bayes, maximum entropy, and SVM models to Turkish political news, leveraging character-level n-grams [6], [18]. Notably, [6] extended the framework in [19] by integrating transfer learning. Despite these advances, traditional machine learning and lexicon-based approaches—such as those presented in [10], [20]-[22]—have generally fallen short of achieving performance levels adequate for real-world deployment. Recent neuro-evolution pipelines can auto-design entire NLP architectures [8]. Miikkulainen *et al.* [8] emphasize the optimization of the complexity-performance trade-off, a principle that guided our Fast CNN design. On the other hand, several deep convolutional neural network (ConvNet) models are currently considered state-of-the-art for text classification tasks, particularly SA. For instance, the "Turkish" model [14] was designed specifically for Turkish SA and includes six dense layers with varying output dimensions. It employs 0.5 dropout after the second and fourth layers, and 0.2 dropout after the third. Another model, "One Layer" [23], is named for its single convolutional layer and comprises an embedding layer, a convolutional layer, a max-pooling layer, and a dense layer. The "Sensitivity Analysis" model [19] features three convolutional blocks of filter sizes 2, 3, and 4, each followed by max pooling and a softmax classifier. The "Rand" model [16] is a scaled-down variant of a well-known CNN, with reduced filter counts and layer dimensions. It includes two convolutional-maxpool-flatten blocks followed by dense layers with dropout. The very deep convolutional neural network (VDCNN) [24] architecture comprises 29 convolutional blocks, each consisting of two convolutional layers, batch normalization, and ReLU activation. CharCNN-Zhang [25], also known as the 9-layer model, comprises six convolutional layers followed by three fully connected layers. Finally, CharTCN [26] utilizes temporal convolution to maintain sequence length, employing a fully convolutional structure that is well-suited for sequential text processing. Similar to [14]-[17], our approach is developed explicitly for Turkish. However, unlike these models, our architecture was derived from extensive experimentation with various layers to identify the most efficient configuration. The models discussed above typically achieve accuracies around 70%, whereas our proposed system demonstrates significantly higher performance across all benchmark datasets. Moreover, our model—like those in [15], [23]—is compact in terms of depth, yet it exhibits notably superior efficiency and performance. As our experiments confirm, deeper architectures, such as those in [24]-[26], do not effectively handle Turkish morphology, resulting in lower classification accuracy.

3. OVERVIEW OF THE PROPOSED OPINION MINING SYSTEM

This section presents the proposed system, along with its various components. The main components are the data gathering stage, pre-processing, feature extraction, and classification modules.

3.1. Data gathering

The following four datasets are used to test the proposed method. The Turkish SA Dataset [27] contains reviews about movies and hotels that were collected from two popular movie and hotel recommendation websites, i.e., "beyazperde.com" and "otelpuan.com." Related to the movie reviews, it contains 53,400 reviews, each of which has already been rated by its author using a scale of 1 to 5 stars. Here, reviews rated 1 or 2 stars are considered negative, those rated 3 stars are considered natural, and those rated 4 or 5 stars are considered positive. Regarding the hotel reviews, 5,802 negative reviews with ratings of 0 to 40 and 5,800 positive reviews with ratings of 80 to 100 were selected. The 17.000 Turkish Tweet dataset [28] was created by collecting the users' feedback about Telecom companies in Turkey. 17,289 Twitter (X) messages were collected and classified as positive, negative, or natural. The Turkish microblog Dataset (TWT) [29] was created using the Twitter API by writing some positive and negative queries and collecting the retrieved microblogs. Overall, this dataset contains approximately 16,400 negative and 16,000 positive microblog tweets. The Turkish movie reviews dataset [30] is similar to the first dataset and related to Turkish movie reviews. However, another group of researchers has collected a different set of reviews consisting of 34,990 positive and negative Turkish movie reviews. It is worth noting that, although the first and last datasets were collected from the same website, we found that only 5,000 reviews are common to both datasets.

3.2. Data pre-processing

Data quality is critical for developing an efficient system, as noisy and poorly prepared data can significantly impact overall performance. The primary objective of preprocessing is to improve the quality of the data samples and eliminate unnecessary information. The developed preprocessing model consists of normalization, data cleaning (including removal of special characters and stop words), correction of typos and misspelled words, and stemming. The normalization process applied standard case folding and tokenization to unify the input format and segment the text [31]. Data cleaning removed non-informative elements, such as punctuation, stop words, URLs, and special characters. Correcting typos and misspelled words is essential, as these errors can alter the meaning of an entire sentence. De-ASCIIification was applied to restore Turkish-specific characters from ASCII representations. Stemming refers to the process of identifying the root of each word. The Turkish language has a rich morphological structure, enabling the creation of numerous new words by adding a variety of suffixes to a single stem—for instance, the verb "anlamak" (meaning "to understand"). By adding the suffix "-t" to the verb stem "anla-", we form "anlat-(mak)", which translates to "to tell". When we further add the suffix "-l", it becomes "anlatl-(mak)", indicating "being told". Turkish has around fifty suffixes that can be attached to verbs, and multiple suffixes can be used simultaneously. Turkish verbs often contain multiple suffixes encoding tense, negation, and subject. For example, kalamyormuşsunuz contains five suffixes built on the root kal- (meaning "to stay").

3.3. Feature extraction

At this stage, we convert the text into a real-valued vector using the vector space model (VSM) [32]. This algebraic framework transforms the text into numerical vectors. In more detail, the samples of the used datasets are the input, and each sample is represented by a feature vector of its words; the features of a word (F_w) can be represented by:

$$F_w = \left[\, f_1, f_2, f_3, \ldots, f_M \,\right]^{\mathsf{T}} \tag{1}$$

Where f is a floating-point number produced by the chosen method and M is the length of the vector. For example, using TF–IDF, M represents the total number of samples in the utilized dataset, and the TF–IDF vector of a word represents that word across the N documents in the current dataset; hence $M = N$. However, we may use optimization algorithms—such as `GridSearchCV` from `scikit-learn`—to select a subset of the dataset when calculating the TF–IDF for the i-th word; in this case, $M \leq N$. Next, each word is represented by the average of its M floating-point values $\left(\overline{F}_w\right)$, and each sample is represented by the features of 150 words, i.e., as shown in (2), the features of the sample are obtained by concatenating these averages.

$$F_s = \left[\, \overline{F}_{w_1}, \overline{F}_{w_2}, \overline{F}_{w_3}, \ldots, \overline{F}_{w_{150}} \,\right]^{\mathsf{T}} \tag{2}$$

We set the sequence length to 150 tokens, selected based on the average document length in our datasets and confirmed in preliminary experiments using term frequency-inverse document frequency (TF-IDF), word to vector (Word2Vec), bidirectional encoder representations from transformers (BERT), and eXtreme language net (XLNet). Overall, the features of the used dataset (F_D) are represented by (3):

$$F_D = \begin{bmatrix} F_{s_1}^{\mathsf{T}} \\ F_{s_2}^{\mathsf{T}} \\ \vdots \\ F_{s_N}^{\mathsf{T}} \end{bmatrix} \tag{3}$$

Hence, the result of this step is a vector of size (N, 150). As illustrated in Figure 1, this study investigated traditional Bag-of-Words approaches, such as TF-IDF, and cutting-edge word embedding algorithms, including Word2Vec, BERT, and XLNet. These techniques were explicitly explored, and their performance in processing Turkish was thoroughly investigated in this study.

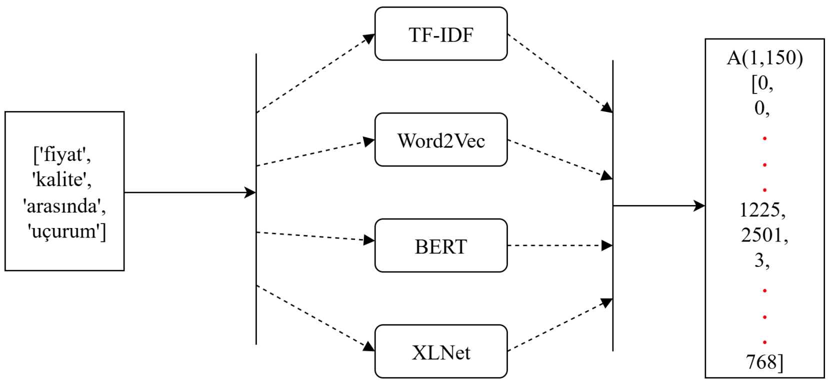

Figure 1. Feature extraction algorithms in the proposed SA system. Dashed arrows denote the optional selection of one among TF-IDF, Word2Vec, BERT, or XLNet

These strategies are briefly outlined below:

- TF-IDF: consists of TF, which indicates the frequency of terms within a text. Notably, the TF approach assigns the highest value to the terms that occur most frequently. The IDF, on the other hand, assesses the significance of a phrase. As a result, the most commonly used terms are not generally the most important. Thus, in IDF, rare words receive the maximum value, whereas frequent words receive a low score. TF-IDF was used as a baseline vectorization method, encoding term frequency and rarity. The TF-IDF is determined by multiplying the TF and IDF values.

$$\mathrm{TF-IDF}(w_i, d) = \frac{O(w_i, d)}{\sum_j O(w_j, d)} \cdot \log\left(\frac{N}{\mathrm{df}(w_i)}\right), \tag{4}$$

In the VSM, $O(w_i, d)$ is the count of term w_i in document d, $\sum_j O(w_j, d)$ is the length of d, N is the number of documents in the corpus, and $\mathrm{df}(w_i)$ is the number of documents containing w_i.

- Word2Vec: can be implemented using the Skip-Gram and continuous bag-of-words (CBOW) models to learn word embeddings from contextual co-occurrence. The first method summarizes the words in the input sentence to estimate the neighbouring words. On the other hand, the second method takes the context of each word as input and attempts to predict the word that corresponds to that context.
- BERT: is a language representation model that leverages deep bidirectional encoding of text sequences. It produces context-sensitive embeddings by considering both the left and right context of a word simultaneously. For example, the word "running" will have distinct vector representations in the sentences "He is running a company" and "He is running a marathon", reflecting different meanings.
- XLNet: is a permutation-based autoregressive model. During pre-training, each token is predicted from a subset of tokens that precede it in a randomly generated permutation, allowing the network to leverage both left- and right-hand context across training passes. We therefore included XLNet embeddings as an advanced baseline. In this study, we utilized the publicly available BERT and XLNet embeddings, which were fine-tuned on the 2024 Turkish Wikipedia dump and hosted via Hugging Face and TensorFlow repositories [13], [33]. These embeddings capture the morphological and syntactic nuances of Turkish more effectively than generic models, thereby improving downstream classification performance.

3.4. Classification

A new system for efficiently handling Turkish has been introduced. This system was developed after extensive research to determine the optimal layers, including embedding, global max pooling, dropout, dense, convolution, long short-term memory (LSTM), and gated recurrent unit (GRU). Figure 2 illustrates the proposed CNN architecture. The suggested architecture comprises several layers: an embedding layer, a global max pooling layer, and two dense (fully connected) layers.

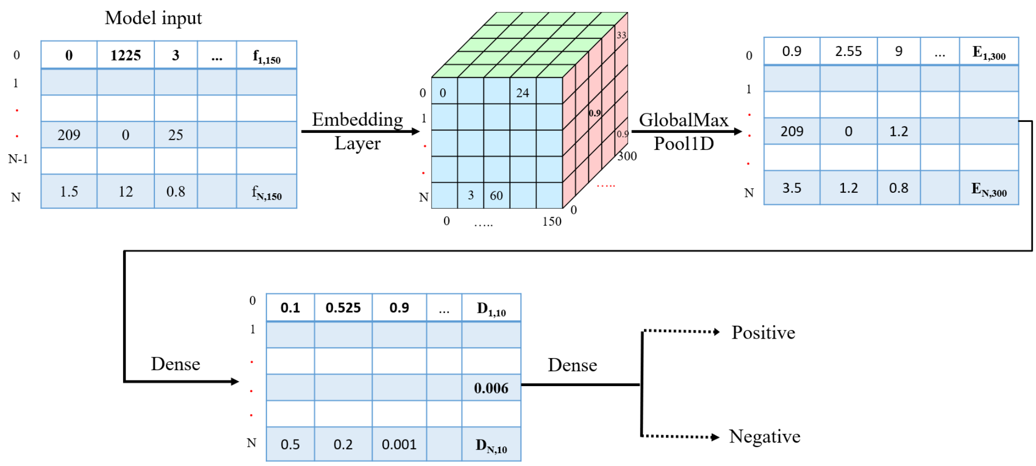

Figure 2. An approximate representation of the constructed CNN architecture

Below is more information about the essential components (layers) of the constructed CNN model:

- Embedding layer: this layer is designed for NLP applications. It effectively preserves contextual similarity among words; that is, representations of words with comparable meanings are close in the embedding space. The proposed approach begins with this layer. The input to the layer is the tokenized sequence of each document as integer indices, i.e., $F_D \in \mathbb{N}^{N\times 150}$. The embedding layer comprises 300 dimensions; its weight matrix $W \in \mathbb{R}^{V\times 300}$ (with V the vocabulary size) maps each token to a 300-dimensional vector, producing the embedding output $E_D \in \mathbb{R}^{N\times 150\times 300}$. Equivalently, for each sample, the feature extraction step yields a vector of size $(150, 1)$, while the embedding matrix (the layer weights) has size $(300, 150)$. In this setting, 300 neurons are used and 150 denotes the document-vector length. Multiplying these yields an embedding output $E_D \in \mathbb{R}^{N\times 150\times 300}$.

- Pooling layer: pooling is the second most commonly used layer, focusing on down-sampling feature maps to reduce the size of the activation map. Given the input $E_D \in \mathbb{R}^{N\times150\times300}$, a global max 1D pooling layer reduces the representation to $\mathbb{R}^{N\times300}$; we refer to this output as EmaxD.
- Dropout layer: during training, this layer randomly disregards some neurons and their corresponding incoming and outgoing connections—that is, specific neurons are temporarily removed for each training sample—to mitigate overfitting. The proposed CNN applies dropout with a rate of 0.2 after the global max-pooling layer and between the two dense layers.
- Dense layer: also known as a fully connected layer, this type of layer connects every neuron to those in the subsequent layer. The proposed approach uses two dense layers: the first maps an input of size $[N, 300]$ to an output of size $[N, 10]$, and the second produces a single output per sample, identifying each sample's class as either positive or negative.

4. EXPERIMENTS

This section presents several experiments to examine the proposed system's performance supporting the Turkish language. Various datasets were utilized in this study to ensure the robustness of the developed system. The details of these datasets are presented in the "Data Gathering" section. We report the area under the curve (AUC), computed via the trapezoidal rule.

4.1. Experiment setups

The developed method was implemented using Python and various libraries, including Keras, Pandas, and NumPy. For De-Asciification, we utilized the Turkish-Deasciifier library referenced in [13], while text normalization was implemented using Turkishnlp [33], which was updated in 2024 with an expanded vocabulary and improved Unicode handling. The stemming was performed using the Python version [34] of the TurkishStemmer introduced in [35]. Additionally, we imported functions for preprocessing and feature extraction from Keras and Scikit-learn. We used the "Bert-for-tf2" and "Bert base multilingual uncased" models for the word embeddings layer. The "bert-for-tf2" library is the TensorFlow BERT tokenizer implementation. The "Bert base multilingual uncased" is available in 102 languages, including Turkish. In the case of Elmo, we used the "TensorFlow Hub" and the most recent version of its model. Finally, to retrain the XLNet, the "transformers" library was used to import XLNetConfig, XLNetForSequenceClassification, XLNetTokenizer, and XLNetModel. It's worth noting that the used embedding systems were trained first using the "Turkish Wikipedia dump" dataset. In contrast, the Turkish pre-trained word vector was trained using the "Turkish CoNLL17 corpus. Overall, we prefer the latter due to its superior performance.

4.2. Experiments and results analysis

4.2.1. Experiment #1: comparing our approach with the state-of-the-art CNN models

In this experiment, we compared the performance of our proposed Fast CNN model against seven well-known CNN-based text classification architectures: Turkish [15], One Layer [23], SensitivityAnalysis [19], Rand [16], VDCNN [24], CharCNNZhang [25], and CharTCN [26]. For fairness, all models were reimplemented using their original hyperparameter settings as described in their respective publications. To ensure consistent feature input, we used BERT-based embeddings across all models, given its superior performance relative to other feature extraction methods, as established in Experiment #2.

As illustrated in Figure 3, our Fast CNN outperforms all seven competing CNN-based models. It achieves an average AUC of 0.94, a 6.8% improvement over the strongest baseline (Rand). Notably, deep models such as VDCNN and CharTCN exhibit lower performance ($\mathrm{AUC} \approx 0.50$–$0.52$), likely due to their inability to generalize to the agglutinative morphology of Turkish. Interestingly, the Rand model performed better than reported in [16], a result we attribute to replacing its original feature extractor with BERT. These results demonstrate that our lightweight architecture not only simplifies computation but also generalizes more effectively across diverse Turkish-language datasets than deeper, more resource-intensive models.

4.2.2. Experiment #2: a comparison of feature extraction techniques' performance

In this experiment, we examined the performance of the commonly used traditional feature extraction approach, TF–IDF, alongside several cutting-edge word-embedding methods, including BERT, XLNet, FastText, and Word2Vec. More specifically, each feature-extraction technique was integrated into the proposed model and evaluated to identify the method best suited for Turkish. Furthermore, FastText was implemented using both bigrams and trigrams. As shown in Figure 4, BERT embeddings consistently yield the highest AUC across all datasets, reflecting their ability to capture Turkish linguistic nuances. Word2Vec follows, while traditional TF–IDF performs the weakest, reinforcing the superiority of contextual embeddings for languages with rich morphology.

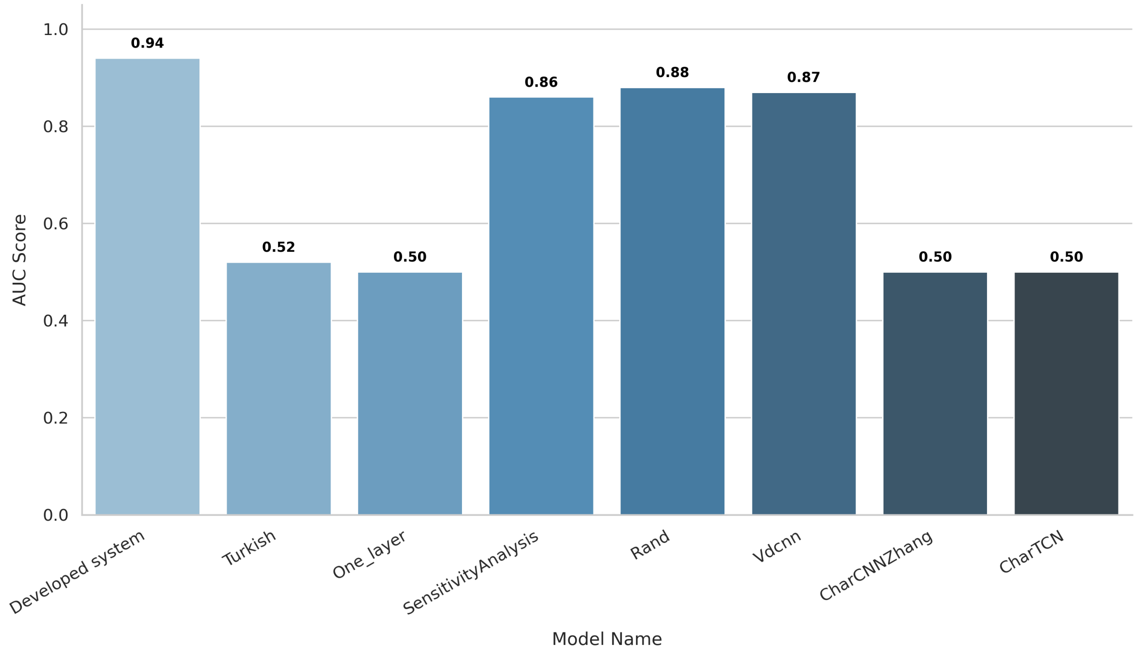

Figure 3. Average AUC across four Turkish datasets for eight classifiers, with Fast CNN leading at 0.94—6.8% above the next-best (Rand)

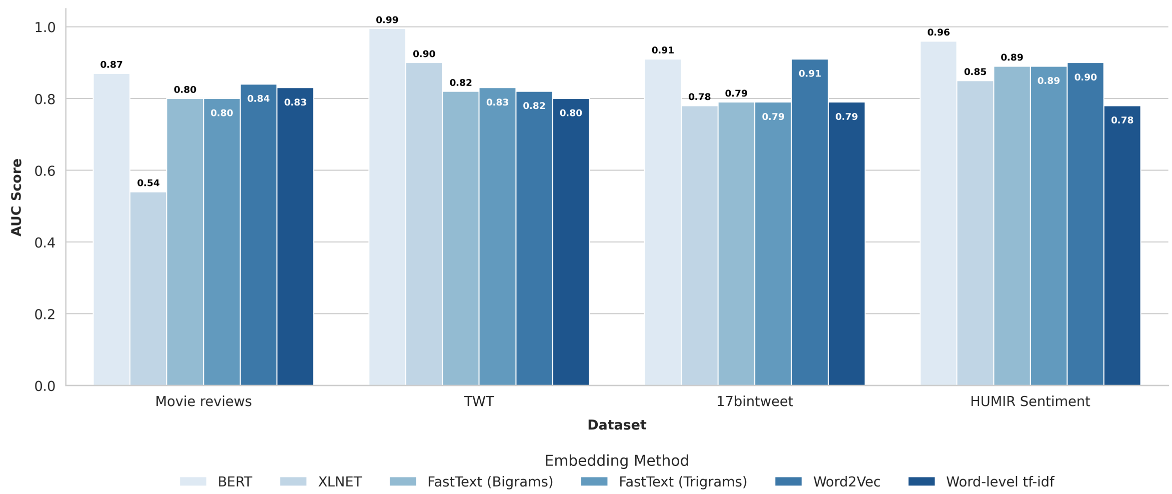

Figure 4. AUC comparison of TF–IDF, Word2Vec, FastText, BERT, and XLNet across four Turkish datasets, where BERT yields the best overall performance

4.2.3. Experiment #3: text pre-processing

This experiment aims to assess the impact and necessity of each preprocessing step, explained in the "Data Pre-processing" section, in dealing with the Turkish language. Additionally, we have investigated the effect of applying all preprocessing steps, both with and without the stemmer. Therefore, this experiment can help determine the extent to which performance can be influenced by either pre-processing the data or using it directly. Given the limitations of the number of pages, we calculated the average AUC score improvement across all studied models.

As seen in Table 2, stemming negatively impacts classification accuracy—especially on morphologically rich Turkish—by distorting root meanings. Conversely, De-ASCIIification and noise elimination lead to consistent AUC improvements. Preprocessing without stemming yields the highest gains, highlighting the importance of preserving morphological structure.

Table 2. AUC improvements (%) across four datasets for individual preprocessing steps. The best performance is achieved when stemming is excluded

Preprocessing step	Movie reviews	Twt	17BinTweet	HUMIR
De-Asciification	0.554151855	0.1187543	2.071717	0.181047
Doc correct	0.339610423	0.017390736	1.144129	0.280983
Eliminating useless info	1.33662074	0.755627262	0.602438	2.895684
Stemmer	-0.724926814	-0.36847528	0.389827	2.969275
Pre-processed-with-stemmer	-1.179471608	-1.10078906	1.523033	3.913083
Pre-processed-without-stemmer	4.73989231	3.910404629	4.972882	7.551489

4.3. Discussion and cross-language applicability

Although this study focuses on Turkish, the proposed Fast CNN model is designed to be language-agnostic and can be adapted to other agglutinative languages such as Korean, Japanese, and Finnish. These languages share similar morphological complexities—particularly extensive suffixation—that challenge traditional SA techniques. The use of contextual embeddings and a compact CNN architecture in our system enables effective handling of such linguistic structures. While additional experiments on other languages are beyond the scope of this paper, future work will explore extending the model to other typologically similar languages. The model's lightweight design and proven accuracy make it a strong candidate for multilingual or resource-constrained SA applications. The model's strong performance and compact architecture make it suitable for deployment in real-world Turkish-language systems, such as customer review monitoring platforms, telecom sentiment dashboards, e-government feedback analytics, or news sentiment tracking for media agencies.

5. CONCLUSION

This paper introduces a fast and effective Fast CNN model tailored explicitly for Turkish SA. Addressing challenges posed by Turkish's agglutinative morphology, our approach integrates a compact architecture with advanced contextual embeddings. Experimental results across four Turkish sentiment datasets demonstrate that the proposed model achieves an average AUC of 0.94, outperforming seven state-of-the-art CNN-based models by 6.8% over the strongest baseline and by over 80% relative to several deeper architectures. Compared to conventional and deeper CNN models, our method offers both higher accuracy and lower computational complexity, making it well-suited for real-world applications such as social media monitoring, product review mining, and customer opinion analysis in Turkish.

Future research will involve extending the model to other morphologically rich languages, evaluating its effectiveness in multilingual and cross-lingual settings, and exploring lightweight deployment in mobile or edge environments.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Saed Alqaraleh	✓	✓	✓		✓	✓		✓	✓	✓				✓
Abdul Hafiz AbdulHafiz		✓		✓	✓	✓		✓	✓	✓	✓	✓		

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

All datasets analyzed in this study are publicly available online from their original sources. Direct access to information and citations is provided in the *Data Gathering* subsection, along with references [4], [6], [14], [31]. No new datasets were generated by the authors.

REFERENCES

[1] R. Dehkharghani, Y. Saygin, B. Yanikoglu, and K. Oflazer, "Sentiturknet: A turkish polarity lexicon for sentiment analysis," *Language Resources and Evaluation*, vol. 50, no. 3, pp. 667–685, 2016, doi: 10.1007/s10579-015-9307-6.

[2] S. Sunkapaka and N. V., "Sentiment analysis using Telugu SentiWordNet," in *2023 2nd International Conference on Futuristic Technologies (INCOFT)*,Belagavi, Karnataka, India, 2023, pp. 1-5, doi: 10.1109/INCOFT60753.2023.10425658.

[3] A. AbdulHafiz, "Novel opinion mining system for movie reviews," *International Journal of Information Engineering and Electronic Business*, vol. 8, no. 2, pp. 94–101, 2020, doi: 10.18201/ijisae.2020261590.

[4] S. Alqaraleh, "Turkish sentiment analysis system via ensemble learning," *Avrupa Bilim ve Teknoloji Dergisi*, pp. 122–129, Aug. 2020, special issue, doi: 10.31590/ejosat.779181.

[5] M. Gardner *et al.*, "Allennlp: A deep semantic natural language processing platform," in *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, Melbourne, Australia, Jul. 2018, pp. 1–6, doi: 10.18653/v1/W18-2501.

[6] M. Kaya, "Sentiment analysis of turkish political columns with transfer learning," Master's thesis, Middle East Technical University, Ankara, Turkey, 2013.

[7] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017, doi: 10.1109/MIS.2017.23.

[8] R. Miikkulainen *et al.*, *Evolving deep neural networks*, Artificial Intelligence in the Age of Neural Networks and Brain Computing, Ed. Cambridge, MA, USA: Academic Press, pp. 293-312, 2019, doi: 10.1016/B978-0-12-815480-9.00015-3.

[9] M. Okada, H. Yanagimoto, and K. Hashimoto, "Sentiment classification with gated CNN for customer reviews," in *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Pattaya, Thailand, 2018, pp. 1-5, doi: 10.1109/iSAI-NLP.2018.8692959.

[10] M. A. Tocoglu, O. Ozturkmenoglu and A. Alpkocak, "Emotion analysis from Turkish tweets using deep neural networks," *IEEE Access*, vol. 7, pp. 183061-183069, 2019, doi: 10.1109/ACCESS.2019.2960113.

[11] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," *arXiv preprint*, Feb. 2017, doi: 10.48550/arXiv.1702.0192.

[12] C. Türkmenoglu and A. C. Tantug, "Sentiment analysis in Turkish media," in *International Conference on Machine Learning (ICML)*, Jun. 2014, pp. 1–8.

[13] K. Adali and G. Eryiğit, "Vowel and diacritic restoration for social-media texts," in *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, Gothenburg, Sweden, Apr. 2014, pp. 53–61, doi: 10.3115/v1/W14-1307.

[14] B. Ciftci and M. S. Apaydi, "A deep-learning approach to sentiment analysis in turkish," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Turkey, 2018, pp. 1-5, doi: 10.1109/IDAP.2018.8620751.

[15] G. M. Demirci, Ş. R. Keskin and G. Doğan, "Sentiment analysis in turkish with deep learning," in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 2215-2221, doi: 10.1109/BigData47090.2019.9006066.

[16] F. Kurt, "Investigating the performance of segmentation methods with deep learning models for sentiment analysis on Turkish informal texts," Master's thesis, Middle East Technical University, Ankara, Turkey, 2018.

[17] S. Yildiri, *Comparing deep neural networks to traditional models for sentiment analysis in Turkish*, Deep Learning-Based Approaches for Sentiment Analysis. Cham, Switzerland: Springer, pp. 311–319, 2020, doi: 10.1007/978-981-15-1216-2_12.

[18] M. Kaya, G. Fidan, and I. H. Toroslu, "Sentiment analysis of Turkish political news," in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Macau, China, Dec. 2012, pp. 174–180, doi: 10.1109/WI-IAT.2012.115.

[19] Y. Zhang and B. M.Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint*, Oct. 2015, doi: 10.48550/arXiv.1510.03820.

[20] R. Dehkharghani, B. Yanikoglu, Y. Saygin, and K. Oflazer, “Sentiment analysis in turkish at different granularity levels,” *Natural Language Engineering*, vol. 23, no. 4, pp. 535–559, 2017, doi: 10.1017/S1351324916000309.

[21] V. K. Singh, R. Piryani, A. Uddin, P. Waila, and Marisha, “Sentiment analysis of textual reviews: Evaluating machine-learning, unsupervised and Senti- WordNet approaches,” in *2013 5th International Conference on Knowledge and Smart Technology (KST)*, Chonburi, Thailand, 2013, pp. 122-127, doi: 10.1109/KST.2013.6512800.

[22] S. Tofighy and M. F. Fathian, “A proposed scheme for sentiment analysis: Effective feature reduction based on statistical information of SentiWordNet,” *Kybernetes*, vol. 47, no. 5, pp. 957–984, 2018, doi: 10.1108/K-06-2017-0229.

[23] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint*, Aug. 2014, doi: 10.48550/arXiv.1408.5882.

[24] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun, “Very deep convolutional networks for text classification,” *arXiv preprint*, Jun. 2016, doi: 10.48550/arXiv.1606.01781.

[25] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *NIPS’15: Proceedings of the 29th International Conference on Neural Information Processing Systems*, Dec. 2015, vol. 1. pp. 649–657.

[26] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint*, Mar. 2018, doi: 10.48550/arXiv.1803.01271.

[27] E. Baki, “Turkish-movie-review-ranking-and-sentiment-predictions,” [Online]. Available: https://github.com/lemogras/, Jul. 2024, gitHub repository; published 2024-07-07. (Date accessed: Jul. 09, 2025).

[28] M. F. Amasyali, H. Tasköprü and K. Çaliskan, “Words, meanings, characters in sentiment analysis,” in *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Adana, Turkey, 2018, pp. 1-6, doi: 10.1109/ASYU.2018.8554037.

[29] A. Hayran and M. Soysal, “Sentiment analysis on microblog data based on word embedding and fusion techniques,” in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, Turkey, 2017, pp. 1-4, doi: 10.1109/SIU.2017.7960519.

[30] A. Ucan, B. Naderalvojoud, E. A. Sezer, and H. Sever, “Sentiwordnet for a new language: Automatic translation approach,” in *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Naples, Italy, 2016, pp. 308-315, doi: 10.1109/SITIS.2016.57.

[31] M. Turker, M. E. Ari, and A. Han, “VNLP: Turkish NLP package,” *arXiv preprint*, Mar. 2024, doi: 10.48550/arXiv.2403.01309.

[32] N. Tas, “RoBERTurk: Adjusting RoBERTa for Turkish,” *arXiv preprint*, Jan. 2024, doi: 10.48550/arXiv.2401.03515.

[33] C. Metehan, “turkishnlp,” [Online]. Available: https://pypi.org/project/turkishnlp/, Jul. 2024, pyPI project; published 2024-07-01. (Date accessed: Jul. 09, 2025).

[34] T. Osman, “Turkish language stemmer for Python,” [Online]. Available: https://github.com/otuncelli, Jul. 2024, GitHub repository; published 2024-07-01. (Date accessed: Jul. 09, 2025).

[35] G Eryiğit and E Adalı, “An affix-stripping morphological analyzer for Turkish,” in *Proceedings of the IASTED International Conference Artificial Intelligence and Applications*, Innsbruck, Austria, Feb. 2004, pp. 299–304.

BIOGRAPHIES OF AUTHORS

Saed Alqaraleh is an Assistant Professor in the Department of Data Science and Artificial Intelligence, College of Information Technology, Mutah University, Karak, Jordan. He received a B.Sc. in Software Engineering from Al-Hussein Bin Talal University, Jordan, and M.Sc. and Ph.D. degrees in Computer Engineering (Artificial Intelligence) from Eastern Mediterranean University, Cyprus. His research interests include visual place recognition, natural language processing (NLP), sentiment analysis (SA), information extraction, data mining, information retrieval systems, search engines, and web crawlers. In this study, he contributed to the problem formulation, experimental design, and interpretation of results for Turkish sentiment analysis. He is active in curriculum development and supervision in data science and AI, and collaborates with regional partners on applied machine learning projects. He can be contacted at email: saed.alqaraleh@mutah.edu.jo.

Abdul Hafiz AbdulHafiz is an Associate Professor at King Faisal University, Al-Ahsa, Saudi Arabia. He received an M.Tech. in Artificial Intelligence from the University of Hyderabad in 2003 and a Ph.D. in Computer Science and Engineering from Osmania University in 2008, in collaboration with IIIT-Hyderabad. A Fellow of the Higher Education Academy (FHEA), UK, his research focuses on robotic vision, machine learning, autonomous driving, and assistive technologies. His work has appeared in leading venues, including ICRA and IROS, and in international journals. In this paper, he contributed to the methodological design and validation of the proposed approach and to manuscript development. He has led multidisciplinary projects that translate intelligent systems into real-world applications and is engaged in academic leadership and research capacity building. He can be contacted at email: aabdulhafiz@kfu.edu.sa.