SportPhraseARabia (SPAR): A Sequential LSTM-based Arabic Auto-Completion System for Sports

Hedaya Mustafa  
*Computer Engineering Department*   
*Birzeit University*Birzeit, Palestine  
1182126@student.birzeit.eduNasser Awwad   
*Computer Engineering Department*   
*Birzeit University*Birzeit, Palestine  
1191484@student.birzeit.eduMazen Batrawi  
*Computer Engineering Department*   
*Birzeit University*Birzeit, Palestine  
1190102@student.birzeit.edu

*Abstract*— Autocompletion systems are currently important for improving the user experience and efficiency in a wide range of applications. This study offers SportPhraseARabia (SPAR) a system for autocompletion designed specifically for Arabic sports articles. SPAR predicts the next word or phrase based on the user's input using a sequential LSTM (Long Short-Term Memory) neural network model.

Keywords— SPAR, autocompletion system, LSTM, sequential model, Arabic language, SANAD, sports articles, prediction.

# Introduction

Autocomplete systems have more popularity in recent years, providing improved user experiences and greater efficiency across a wide range of applications. These systems utilize predictive models to suggest the next word or phrase based on user input, improving typing speed and accuracy. In this study, we present our efforts in developing an auto-completion system called SPAR specifically tailored to assist users in typing sports-related phrases or sentences in Arabic.

NLP has made great progress, particularly in tasks involving language generation. Autocomplete systems play an important role in assisting users during text input by giving contextually relevant real-time suggestions. Our focus on sports-related information addresses the specific needs of users who require assistance with typing sports-related words or sentences.

To construct SPAR, we employed a sequential LSTM neural network model. LSTMs, a type of recurrent neural network (RNN), excel in capturing sequential dependencies and long-term contextual information. By training the LSTM model on a dataset comprising sports-related articles, we aim to empower the system to generate accurate and contextually appropriate autocompletion in the field of sports.

Our work revolves around the utilization of the SANAD (Single-label Arabic News Articles Dataset), a comprehensive dataset sourced from reputable Arabic news outlets. The SANAD dataset encompasses a vast collection of articles, with a specific emphasis on Modern Standard Arabic (MSA) content. We repurposed this dataset, originally intended for text classification tasks, to train our auto-completion system, leveraging its diverse sports category.

In this study, we delve into the methodology employed to preprocess and tokenize the SANAD dataset, rendering it suitable for training our LSTM model. We provide an in-depth understanding of the architecture of SPAR, highlighting the implementation of an embedding layer for capturing semantic representations, LSTM layers for sequence learning, and a dense layer with SoftMax activation for prediction. Furthermore, we explore model optimization strategies employed to enhance the accuracy and relevance of autocompletion predictions, such as the sparse categorical cross-entropy loss function and the Adam optimizer. Additionally, we evaluate the system's accuracy and efficiency.

The contributions of this study encompass the development of an auto-completion system tailored for sports-related text input in Arabic, employing an LSTM-based approach. We address the challenges and potential issues associated with utilizing the SANAD dataset. Our findings not only showcase the potential applications of autocompletion systems in the realm of sports but also contribute to the field of Arabic NLP research.

The remainder of the paper is structured as follows: The second section provides an extensive review of relevant work in autocompletion systems and LSTM-based models. The third section outlines the methodology utilized for data preprocessing, model architecture, and training procedures in detail. Section IV presents the research design, performance analysis, and results evaluation. Section V discusses results of our auto-completion system. Finally, the sixth section concludes the paper, highlighting avenues for future enhancements and extensions of our work.

# Review of Literature

In the field of NLP, autocompletion systems have received a lot of interest, with many studies focusing on creating effective models and algorithms to improve the user's typing experience. In this section, we review significant literature relative to autocompletion systems and LSTM-based models, highlighting their contributions and developments.

Numerous methods and models have been developed to enhance typing experiences using auto-completion systems, which have received extensive study. Due to its capacity to record sequential dependencies and long-term contextual information, researchers have investigated the use of recurrent neural networks (RNNs), notably LSTM models. The idea of sequence-to-sequence learning with neural networks was first proposed by Sutskever et al. (2014), which cleared the way for LSTM-based auto-completion models.

The study of Arabic natural language processing (NLP) has concentrated on tasks like sentiment analysis and text classification. However, there hasn't been much study done specifically on Arabic sports auto-completion systems. This emphasizes how critical it is to create specialized systems like SPAR (SportPhraseARabia) to meet the unique requirements of Arabic-speaking users while typing sports-related information.

For the development of SPAR, existing auto-completion methods in many fields can offer helpful insights. For instance, research have used support vector machine regression and artificial neural networks for auto-completion. Artificial neural networks were suggested by Yan et al. and Amer et al. for use in auto-completion to predict the next word or sentence. A support vector machine-based intelligent prediction system was created by Song et al. These methods show the promise of using intelligent algorithms for precise auto-completion across a range of fields.

The combination of LSTM models and clever algorithms offers the potential to improve auto-completion systems. Ahmed et al. looked into the viability of employing support vector machines and artificial neural networks for auto-completion prediction. Artificial neural networks were used by Al-Gharbi et al. and Elkatatny et al. to predict drilling fluid parameters in real-time, demonstrating the sturdiness and accuracy of these models. The advantages of introducing intelligent algorithms, such as artificial neural networks, into auto-completion systems are highlighted by these instances.

Additionally, optimization strategies are essential for enhancing the effectiveness of auto-completion systems. In order to train neural network models, a well-liked optimization algorithm called the Adam optimizer has been extensively employed. It is efficient for a variety of tasks, including auto-completion, thanks to its adjustable learning rate and momentum optimization. With the Adam optimizer, SPAR can train more effectively and with better convergence.

# Dataset Description

SANAD is a comprehensive dataset specially developed for text classification tasks in Arabic. The data set is derived from the Arabic media and focuses mostly on MSA.

The dataset includes three popular news websites: Al Khaleej, Al Arabiya, and Akhbarona. it covered seven topics: culture, economics, medicine, politics, religion, sports, and technology.

We trained our model using the "Sports" category for our specific study. By concentrating on this topic, we aimed to provide the model with sample information about sports-related content and enable it to generate accurate and contextually appropriate word predictions within the domain of sports. This targeted training approach enhances the model's ability to effectively auto-complete sports-related words and phrases.

# Methodology

This section describes the methodology we used to create our system. Which contains data collection, preprocessing, model architecture, training, and evaluation.

## Data collection

The SANAD dataset was found from Kaggle. This dataset is a comprehensive collection of news articles gathered from Arabic news portals, specifically focusing on MSA text. A subset of articles contains 1000 file was selected from sports category for our training purposes.

## Data preprocessing

The collected articles went through preprocessing steps to prepare them for training the LSTM model. The preprocessing included tokenization, removing stop words, and stemming. Tokenization involved splitting the text into tokens, enabling the model to understand the semantic meaning of each word. Stop words, such as common prepositions and conjunctions, were removed to reduce noise in the data. Additionally, stemming was applied to reduce words to their root form, further enhancing the model's ability to generalize.

## Model architecture

We built a sequential LSTM neural network model for autocompletion. The model architecture consisted of the following layers:

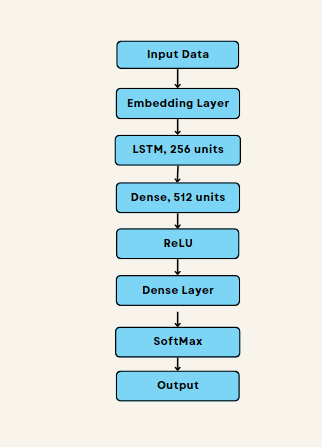
#### Embedding layer: The model began with an embedding layer that captured the semantic representation of words. We used the "Embedding" layer from Keras with a vocabulary size of 35653 and an embedding dimension of 100. The input length was set to 3, indicating that the model processed sequences of three words at a time.

#### LSTM layer: To capture sequential dependencies and patterns in the data, we added an LSTM layer to the model with 256 units

#### Dense layers: Following the LSTM layer, we added a dense layer with 512 units and a ReLU activation function. This dense layer helped the model learn higher-level representations and extract more complex features from the LSTM output. And the ReLU linear function that will output the input directly if it is positive, if not, it will output zero.

#### Output layer: The final layer in the model was a dense layer with 35653 units and a softmax activation function. This layer produced the probability distribution over the vocabulary, indicating the likelihood of each word being the next word in the autocompletion.

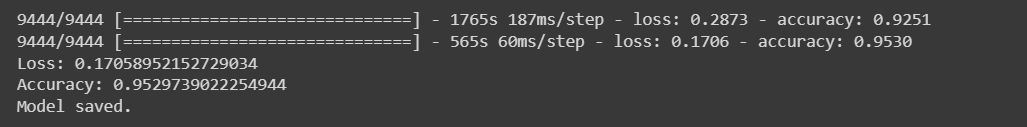
The diagram presented in figure 1 below provides a clear summary of the steps involved in our model.



By using this model architecture, SPAR was able to capture the semantic meaning of words, learn sequential patterns, and generate accurate predictions for the next word or phrase.

#### Model training: The training process involved optimizing the model using the sparse categorical cross-entropy loss function and the Adam optimizer. Adaptive Moment Estimation is a widely-used technique in deep learning to optimize the weight adjustments of a neural network during training. It enhances the traditional stochastic gradient descent method by dynamically assigning a distinct learning rate to each parameter and calculating adaptive estimates for the moments of the first and second gradients. By doing so, the Adam algorithm effectively manages the learning rate throughout the training process, leading to improved convergence and preventing the issues of vanishing or exploding gradients. This optimizer, known as Adam, is especially valuable when training LSTM models that involve intricate and long-term dependencies, as it efficiently handles these complexities. The model was trained for a 100 epochs. We utilized a batch size of 32, which represents the number of training examples processed in each iteration. The choice of batch size was determined based on memory constraints and computational efficiency.

#### Model evaluation: To evaluate the ability of the system to create correct predictions, metrics such as accuracy were computed. We also performed qualitative evaluations by manually examining the suggestions generated by the system for a set of test queries. This qualitative assessment provided insights into the relevance and contextual appropriateness of the generated completions.



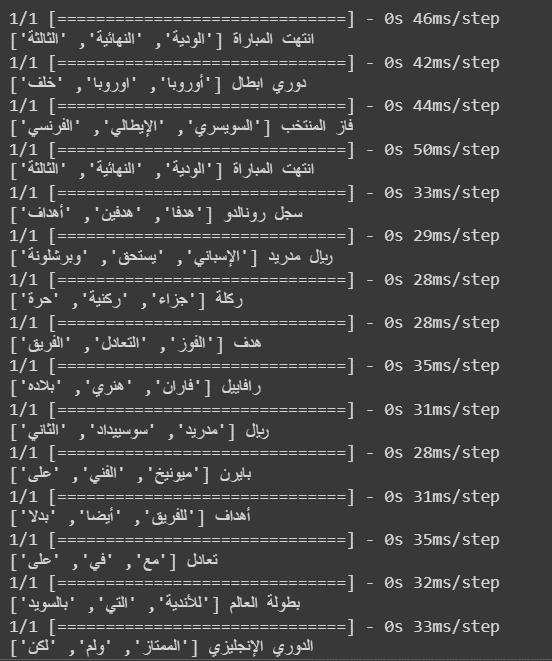
#### Experiment setup: The experiments were conducted on a google colab pro with GPU to ensure sufficient computational resources. In implementing SPAR, the GPU was used because LSTM models are highly computationally intensive, requiring multiple sequential calculations that involve large matrices, vector operations, and parameters. The parallel processing capabilities of GPUs enable the model to perform these operations much faster than CPUs, resulting in significantly reduced training times and increased efficiency. Additionally, the frameworks we used (TensorFlow and Keras), have built-in support for GPU acceleration, making it relatively easy to implement and train LSTM models on GPU-enabled hardware.

#### Hyperparameter tunning: To optimize the model's performance, we made hyperparameter tuning experiments by trial and error method. We explored different hyperparameters, such as the learning rate, batch size, and number of LSTM units, to identify the settings that gave the best results in terms of accuracy and relevance of autocompletion suggestions.

# Results and Discussion

The performance of the SPAR model was evaluated by generating predictions for subsequent words based on a set of input samples.

The analysis of the results, as shown in the following figures, revealed that the model generally provided accurate and contextually appropriate predictions for the majority of the input samples. For example, when given the input phrase "انتهت المباراة" (the match ended), the system correctly suggested words like 'الودية' (friendly), 'النهائية' (final), and 'الثالثة' (third) as the most likely options for the next word, which aligned well with the expected completion.





Additionally, the system consistently generated a varied range of reasonable suggestions. It successfully captured various semantic aspects related to the input samples, including team names, player names, match outcomes, tournament names, and actions associated with football matches. This ability to offer a variety of predictions enhances the system's flexibility and usability for auto-completion.

However, it is important to note that the system occasionally produces less accurate or relevant predictions. In some cases, although the predicted words were grammatically correct, they did not always perfectly match the intended context or the most probable completion. For instance, the system sometimes suggested football team or player names that were not directly relevant to the given input sample.

Overall, the results of this study demonstrate the potential of SPAR to assist users with accurate and relevant word suggestions in Arabic text. Further enhancements and optimizations can be explored to improve its performance, making it applicable across various domains such as text editors, messaging platforms, and language learning tools. By doing so, the system can contribute to enhanced user productivity and an improved user experience.

# Conclusion

This study established the autocompletion system SportPhraseARabia (SPAR), which was developed especially for Arabic sports articles. Based on user input, SPAR uses a sequential LSTM neural network model trained on the SANAD dataset to predict the next word or phrase. The system demonstrated that it was able to capture a range of semantic sports-related aspects and generate precise and contextually appropriate autocompletion suggestions. Despite a few isolated failures, SPAR proved it could enhance the user experience when entering sports-related content. Additional enhancements and optimizations can be looked into to expand its application to other domains, contribute to the field of Arabic natural language processing research, and boost user productivity.

##### References

1. “Word Embeddings  :  Text  :  Tensorflow.” *TensorFlow*, www.tensorflow.org/text/guide/word\_embeddings. Accessed 1 July 2023.
2. Samson, Hasara. “Optimizing Deep Neural Networks through Hyperparameter Tuning.” Medium, 9 Nov. 2020, towardsdatascience.com/optimizing-deep-neural-networks-through-hyperparameter-tuning-1c8ae15bd3c7. Accessed 1 July 2023.
3. Hosni, Youssef. “Building an LSTM Model from Scratch in Python.” *Medium*, 30 Jan. 2023, pub.towardsai.net/building-a-lstm-from-scratch-in-python-1dedd89de8fe. Accessed 2 July 2023.
4. Communicated by Ronald Williams. (n.d.) <https://deeplearning.cs.cmu.edu/S23/document/readings/LSTM.pdf>. Accessed 5 July 7, 2023.
5. Hermessi, Haithem. “Arabic News Articles Dataset.” *Kaggle*, 6 Aug. 2020, [www.kaggle.com/datasets/haithemhermessi/sanad-dataset?select=Sports](http://www.kaggle.com/datasets/haithemhermessi/sanad-dataset?select=Sports). Accessed 22 June 2023.