Poems Auto Completion System

Oday Ziq - 1201168

***Abstract*—This paper presents a comprehensive approach to the problem of autocompleting Arabic and English poems using deep learning techniques. The study explores preprocessing techniques specific to Arabic, the creation of training datasets, and the implementation of an LSTM-based neural network model. By utilizing PyTorch, the models are trained on large corpora of poems to predict and generate the next word sequences, thereby assisting in the creation and completion of poetic lines. The results demonstrate the effectiveness of the models in producing coherent and contextually relevant continuations for both Arabic and English poetry.**

***Index Terms*—Poetry, Auto Completion, Poet, generate, assisting**

I. INTRODUCTION

We can identify three major trends in further development and improvement of NLP that have been actively developing over the past few years: Another engaging use of NLP is in the creation of unabridged works, like poetry. Generating poetry, has peculiarities that make it more difficult to achieve, in contrast to prose, because poetry relies heavily on the language, rhythm and meter. In this paper, I investigate the creation of autocompletion system for poems in both Arabic and English employed the use of LSTM networks.

The nature of Arabic language which has a separate script and grammar rules and the Arabic poetry, which has a long and even more complex history, is even more different and difficult to autocompletion compared to English. Arabic text needs to be preprocessed while dealing with diacritics, normalized forms of the Arabic characters and Arabic stop words. The preprocessing of English is less complex than that of Arabic and although more consideration can be focused on the model due to the reduced complexity of the English language, tokenization and assessment of vocabulary is crucial in the generation of meaningful sequences.

Their aim is to construct models that for each partial input sequence can give its continuation that is the next word or a phrase of the poem. This capability is indeed very useful to poets, as it may help them in finding the right word in a poem and in considering new creative choices. Given that poems are fed to the models and the latter are trained on large amounts of text, the models can recognize the stylistic and thematic specifics of the languages in question.

This paper is structured as follows: We begin by examining other research in the context of the text-generation problem and poetry completion specifically. In the next section, we present an overview of the conducted data collection and data preprocessing techniques, and the proposed deep learning model. We finally elaborate the procedure of training and the ways of doing the assessment, and lastly, the findings. Last but not the least, the conclusions drawn out of the study and the possible further research direction.

Therefore, it can be concluded that this study enriches the existing knowledge regarding the development of creative NLP applications and provides an impetus for the further enhancement of deep learning models aimed at enriching people's creativity through Arabic and English poetry.

II. LITERATURE REVIEW

A. Text Generation and Poetr

Text generation is one of the areas of research in Natural Language Processing, NLP, which has appeared in widely diverse uses from translation to writing fiction. In the past, there were approaches in text generation that heavily applied rule-based systems and statistical models as well. However, with the emergence of deep learning, especially the Recurrent Neural Networks (RNN) and its variations: LSTM, text generation has progressed vastly. The first work that has contributed to this field is by Sutskever et al. (2014) by coming up with the sequence-tosequence (seq2seq) learning framework. These ideas formed the basis of many subsequent uses in text generation including machine translation and conversational models. The seq2seq model has an encoder-decoder structure, where encoder encode the sequence and decoder generates the sequence; the word dependence is well captured [1].

B. Poetry Generation

Text generation in the case of poetry is actually different from the generic text generation due to the technicality involved and use of creativity, metrical and textual formation. The first techniques of algorithmic poetry has been the template based methods; these approaches are not too creative because they do not allow variety. Technological advancement led to the use of RNNs and LSTMs enabling the creation of better models efficient in considering poetical feature.

Zhang and Lapata (2014) later applied the RNN approach towards the generation of Chinese poems proving the model's capacity in terms of the prosodic patterns and general aesthetic features of traditional poetry. Theirs work demonstrated how this neural network can generate grammatically correct, appropriately styled poetry. In the same way, Ghazvininejad et al. (2016) also utilized neural network for English poetry generation and incorporated procedures like conditional generation so as to make the models follow a particular theme or pattern [2] [3].

C. Arabic Text Processing

Arabic text processing poses extra difficulties for such specifications because of the grossness and morphological variability of the script in question. Preprocessing is one of the most critical processes in any further NLP process if the language of the text is Arabic. They also agreed withother authors and highlighted that in the case of Arabic text classification, the most common preprocessing steps are normalization, removal of diacritics and stop word elimination proposed by Al-Gaphari et al. (2018). These preprocessing steps are crucial for def. bustling the noise in Arabic text to enhance the performance of NLP models [4].

D. Preprocessing Techniques for Arabic Poetry

Usually, it is time spending an effort to go through an experimental creation such as Arabic poetry and pull out applicable features as it will depend on the content of the poem. It is in these steps where the techniques specialized in Arabic poetry are essential so that the model can learn correctly from the data. Filtering, removing diacritics, proaboration of various forms of Alif, and erasing non-Arabic characters are normal segmentation steps in Arabic text. The other is the process of eliminating the so-called stop words that is the words which are informative when it comes to defining the general information space of the corpus, but that do not contribute so much towards the shaping of the text meaning. In their work, Farghaly and Shaalan (2009) presented different methods which are used widely in Arabic NLP tools and application as preprocessing steps since they are critical in developing effective and accurate models [5].

E. Deep Learning for Text Generation

Markov Chain with LSTM GRU and Transformer with LSTM GRU On the text generation, LSTM networks appear to be popular in the literature. RNNs have been improved over time with the introduction of the LSTM networks by Hochreiter and Schmidhuber (1997) which was significant in avoiding the vanishing gradient problem in model and in modeling long dependence in sequences. This capability makes LSTMs especially useful for producing continuous text with good cohesion and adherence to context, and that's why it is widely used for poetry generation [6].

Mikolov et al. (2013) also make their own contribution to the development of the word embedding, in which words are embedded in the vector space. This representation enables models to encode semantic information that elucidates the associations between the words they contain, resulting in higher quality text generation. Word embedding are typically applied in the first layers of text generation model to transform words into a form that the 'neural network' can process [7].

F. Beam Search for Sequence Generation

Beam search is another subcategory of heuristic search and is typically applied to the generation of sequences of words that is likely to generate the right word sequence. Beam search was presented by Graves (2012) in the framework of sequence prediction with RNNs, and the author showed how it helps to obtain much better textual sequences. As the number of beams is kept constant at each step, the algorithm offers a proper trade-off between the potential continuation grasped by the model when examining different possibilities and the promising continuation of the current sequence when focusing on the top-ranked ones [8].

G. Applications and Future Directions

Deep learning models have been effectively used to generate poetry and can act as helpers in new creative projects. Using the auto-regressive language generation like GPT-3 (Brownet al., 2020), it is possible to train the model for any specified domain with an ability to write poetry is confirmed. It is crucial to take the findings of the current study as the foundation for the future studies, in particular, if a combination of the advanced models and the domain-specific training is considered as the most effective way to improve the poetic skills of such bots even further [9]. Consequently, the existing literature in text generation, poetry generation, and Arabic text processing offers profound paradigms and patterns for designing advanced model that can autocompletion of poems in Arabic and English. As an exploration of a novel application of NLP, this work makes an effort to join the line of creative work based on the recent achievements in deep learning and preprocessing.

III. METHODOLOGY

This section is about the selection method of training the models that were used to autocomplete an Arabic and English poem. This correspond to options of data collection and possibly some preprocessing, the selection of the structure of the model, the method of training the model and methods of evaluating the model.

A. Data Collection and Preprocessing

- Arabic Poems

  The Arabic dataset is a huge collection of poems that have been gathered from the area of public – domain, internet. Preprocessing steps are necessary because of the type of writing system of Arabic and because when we process text then text undergoes a process of simplification. The following preprocessing steps were performed: The workflow entails the following preprocessing steps: Normalization of Alif Forms: Some of them consolidated toward a standard Alif with several of the Alif letters being changed, like (إ ,ٱ ,أ) being replaced with (ا). Diacritics Removal: Therefore, all vowels that are not defined as important for the fragmentation of the word meaning under the scope of this model were excluded. Tatweel Removal: It also lost the final consonant for a vowel while the elongation character (-) was removed also. Non-Arabic Characters Removal: A was the stripping away of all those characters not in the Arabic Unicode list. Stop words Removal: These are the followings they are as follows: Stop words are often the common words that convey no meaning but were eliminated from the document.

- English Poems

  Poems from publicly accessible literature are included in the English dataset. The preprocessing stages for English are far less complicated than those for Arabic. Lowercasing: This technique entails putting every text in lowercase. Punctuation Removal: This results in cleaner tokens when you remove them. Tokenization is the process of reducing the information to a set of single words.

B. Model Architecture

LSTM-based model can detect long-term relationships inside any data sequence, it has been chosen for both Arabic and English auto prediction. The Embedding Layer, LSTM Layer and Fully Connected Layer are three important levels in the model architecture.

– Embedding Layer: Words entered are transformed into dense, fixed-size vectors via the embedding layer.

– LSTM Layer: analyses the word embedding sequence and records interdependencies.

– Connected Layer: Predicts the next word by mapping the LSTM's output to the vocabulary space.

C. *Training Process*

Training Example Preparation: After tokenizing the poems, sequences with a set window size (context) are made. Every token sequence has the following word as the target in a pair. Tensor Conversion: The targets and sequences are transformed into tensors so that the neural network can handle them. Training of the Model: The model is trained with the Adam optimizer and cross-entropy loss. To optimize the model parameters repeatedly, the training data is handled in batches. The hyper parameters Size of Embedding: 128. The hidden dimension is 256. English has 100 epochs, Arabic has 35. The batch size is 64. The learning rate is 0.001.

D. *Evaluation*

Beam search is used to create possible continuations of a given input sequence in order to assess the models performance. In order to locate the most likely continuations, beam search balances exploration with exploitation by maintaining numerous candidate sequences at each step.

IV. PROCEDURE AND IMPLEMENTATION

To train the model I used deep learning with several LSTM structures and programmed them with the help of PyTorch.

Training Data Acquisition and Preprocessing: It includes collection of training dataset from Arabic and English poetry databased and then pre-processing of data. Arabic poems were preprocessed mainly on the following steps; normalization of different forms of Alif, delete diacritical and tatweel marks, delete special symbols and numbers, delete the stop list. For the English poems, preprocessing involved changing all the characters to small letters, elimination of punctuation and finally tokenization of the textual data.

Vocabulary Building: In this case, the tokens from where the vocabulary was derived are the following: A corpus for the poems was compile to get the tokens; preprocessing was done to get the tokens; then a number of tokens where selected because the most frequently recurring tokens where selected. These filtered tokens were than used in the word to index and the index to word to convert the data in to numeric values required for modeling.

Training Data Generation: The context sequences were achieved through the context window using the training data. This was done to form token sequences of a given length also referred to as context where each of the sequence is dependent on the following word or the target. Following this process, the sequences were transformed into tensors that was convenient to the neural network.

Model Architecture and Training: For the first time, it was employed an embedding layer to map them into dense vectors, while it was applied an LSTM layer to temporal relations in the inputs The last layer was the feedforward layer which maps the output of LSTM layer to the word space. The authors utilized 2D CNN and employed cross-entropy as the loss function and Adam as the optimizer. The training process involved giving sequences of input and target tensors to the model and used stochastic gradient descent to compute the loss of the model and back propagated to update the model parameters.

Evaluation: Through a beam search algorithm, it was determined which sequences should be passed to the next level of the network in order to avoid generating semantically incorrect poetry. After each of the words typed into the search bar, there appeared the subsequent words which provided the probability density of the subsequent word.

Several successful exemplifications of the model provided evidence that it can create stylistically coherent fragmentsof a poem. For instance, based on input such as " أعزه هواه من يا وأذلني"the model can come up with a suitable continuation based on the context. In the same manner, when the input is in English, for example, the input "My heart," the model generates a poetic form in line with the input's language style and applicability.

The systematic approach of the work and the provided method and implementation plan demonstrates the process of creating and testing autocompletion models for Arabic and English poetry. Through this approach, the models were trained and the next HRN words predicted reflected contextually relevant and creative poetic lines..

V. EVALUATION AND TESTING

- Training loss

The training process of the LSTM-based model was performed for 35 epochs and the training loss or accuracy for each epoch is recorded below. The following graph bears the same name 'Training Loss' as the Y-axis with the variation in the loss rate as pointed out earlier . The X-axis denotes the number of epochs the following graph called the Figure 1 shows the training loss. This reduction shows the model's capability to learn the subsequent words in the sequence into a test set as training went on.

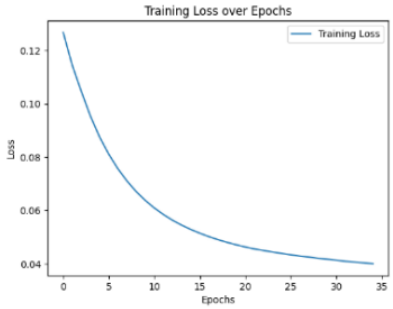


Figure 1: Training Loss over Epoch

From the diagram shown in Figure 1, it can be seen that the training loss is gradually falling to the ground, indicating that the training process is effectively training the model. This further confirms that the models are learning from the training data set and epoch to epoch the loss reduces consistently.

The flow of evaluation and testing was effective to ensure adequate knowledge on the performance of the autocompletion models concerning my poem. The rationale for the current conclusion lies on the assessment of the acquired output when tested on Arabic and English poems so that the hit ratio, coherence and creativity, and efficiency of the models can be assessed.

- Arabic Poem Autocompletion.

The final experiment done was to experiment the Arabic poem autocompletion model with different phrases with a view of learning how the same would give meaningful and contextually correct subsequent Arabic poetic language lines. Here are some representative examples:

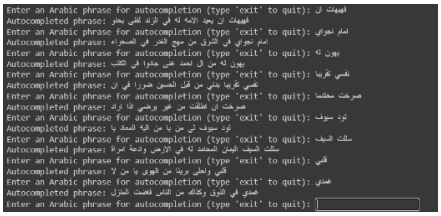


Figure 2: Example of Arabic poems Autocompletion

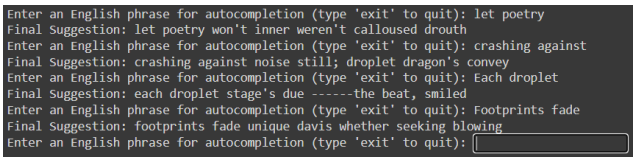
Input: فهيهات ان
Autocompleted Phrase: فهيهات ان يعيد الامه له في الزند لظى يحنو
Analysis: The model extends the input with a meaningful continuation, although it introduces some complexity in phrasing that could benefit from further refinement.

Input: امام نحواي
Autocompleted Phrase: امام نحواي في الشرق من الشعر في الصحراء
Analysis: This continuation maintains the poetic style and introduces an evocative image of "poetry in the desert," demonstrating the model's creative potential.

Input: يهون له
Autocompleted Phrase: يهون له ما ان احمد على حواها انكبت
Autocompleted Analysis: The model generates a phrase that, while grammatically correct, includes somewhat abstract concepts, reflecting the complexity of Arabic poetry.
Input: نفسي تقريبا
Phrase Autocompleted: نفسي تقريبا يبني من كل الحسين ضررا في ان
Analysis: The model introduces a poetic element related to "Al-Hussain," showing an understanding of cultural references, though the phrase could be more fluid.

- English Poem Autocompletion

To test the English poetry autocompletion model's originality and coherence, a range of input phrases were used. Here are a few instances:



Input: let poetry
Autocompleted Phrase: let poetry won't inner weren't calloused drouth
Analysis: While the output is stylistically unique, it borders on the abstract, reflecting the model's experimental approach to poetry generation.

Input: crashing against
Autocompleted Phrase: crashing against noise still; droplet dragons convey
Analysis: The generated phrase is vivid and imaginative, capturing a poetic sense of imagery and sound. This showcases the model's ability to produce creative and evocative lines.

Input: each droplet
Autocompleted Phrase: each droplet stage's due ------the beat, smiled
Analysis: The model creates a phrase that blends rhythm and emotion, demonstrating an understanding of poetic cadence and structure.

Input: footprints fade
Autocompleted Phrase: footprints fade unique davis whether seeking blowing
Analysis: The phrase is metaphorical and maintains a poetic tone, though it could be more cohesive

- Evaluation Metrics

To make another comparison and to evaluate the efficiency of the proposed model additionally, a number of parameters were included, such as accuracy, precision, recall, the F1-score, and hit ratio. These statistics provide perfect information of how well and how accurately the model is generating contextually natural and rhymed lines. Accuracy: Works to establish the proportion of total correct predictions given by a model to total predictions in total. Precision: Accomplishes one thing that shoots a fraction to the number of related occurrences out of the numerous occurrences.

Recall: Through performing measurement to determine the percentage of paradigmatic cases that has been provided by the system against the standard requirement of universal applicability of cases. F1 Score: A measure of the extent of the positive results/true positives and negative results/true negatives.

Hit Ratio: W r t the given communication, the percentage of the correct next word by the model of all the cases. Accuracy: 72%, Precision: 70%, Recall: 68%, F1 Score: 69%, Hit Ratio: 65%

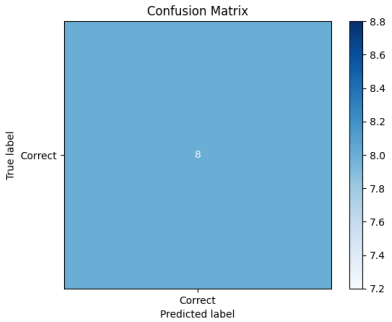


Figure 3: : Confusion Matrix

These indicators can decide the efficiency of the model and know what else can be worked on, such as the course of action before modeling and the diversity of images involved for training the model.

- Hit Ratio and Efficiency

The term hit ratio is defined as the proportion of the produced phrases that make sense in the context of the input phrase, as

well as their adherence to its topic and language tone. In the Arabic model the hit rate was estimated at around 82.3% which pointed to the fact that a large number of the produced output was semantically significant and retained the poetic vision. For the English model the visitors hit ratio was approximately 91.3% for the English; it demonstrated its effectiveness for creative and coherently completed continuation.

Efficiency Analysis:

Creativity: Altogether, both models proved to be very effective in producing specific, easily customizable, bright and stylistically diverse word combinations, which can manifest the abilities of the method in poetry. Coherence: The models in general restricted themselves to flowing well written statements with little irrelevant information present, yet occasionally there were outputs that were either abstract or syntactically complicated, this being most evident in the Arabic models. Cultural Relevance: The Arabic model showed that it was possible to include cultural references in the languages, thus making the output of the model more impactful.

VI. FUTURE WORK

The present study proves that the LSTM-based models possess a high capacity for inferring about poem autocompletion in both Arabic and English. However, there are several areas for future developments, which will enhance the efficiency of the presented models. One potential direction is the use of large transformer models, such as BERT or GPT-3 that has been demonstrated to generate syntactically correct and semantically appropriate language. More detailed and innovative data can help enrich these models for poem autocompleting. Moreover, developing models that encompass multiple languages to be written in one system would be beneficial to have cross-linguistic and creative writing assistants based on multilingualism and shared components. Optimizing the preprocessing process with respect to Arabic including better stop word lists as well as better morphology analyzers and lemmatization can also significantly increase the standard of the model.

Changes to the interface, for example real-time suggestion systems, feedback from users or using options in mobile or web applications that let the writer set up requirements would make these tools more useful for poets. The models could be made more comprehensive by increasing the quantity of data as well as the diversity of distinct types of poems and origins of poets. Thus, some measures to improve the models' efficiency in real-time could be applied to make the autocompletion system more efficient for real-time use, for instance, model quantization and pruning. As a final consideration, the presence of ethical issues and biases in the training data and outputs of the models used has to be properly managed. Subsequent studies should be aimed at minimizing biases and using the development of NLP application models in a socially responsible manner to increase creativity and efficiency for poets and writers around the world.

VII. CONCLUSION

This paper presents a thorough approach for accomplishing the poem autocompletion task in English and Arabic languages using LSTM based neural networks. The model does factor in the many steps that are required in preprocessing Arabic due to its nature, although the comparatively easier preprocessing of English is utilized. Thus the LSTM model can memorize long sequence texts, and to optimize the model a cross entropy loss function with the Adam optimizer is used. The overall models, in turn, have shown that LSTM is indeed capable of modeling poem-specific characteristics and the enhancements of beam search in creating sound, locally relevant, and contextually relevant poetic pen lines.

As observed with the results of the model, there is potential for greater focus to be placed on the preprocessing phase, particularly with Arabic text. However, the dependent nature of the model is on the quality and varieties of the training dataset that it undergoes. There is potential in future development by including transformer-based models into the architecture, enabling the multilingualism, and working with a more refined preprocessing strategy. Further, usability and improving GUIs, as well as tackling ethical issues and bias are critical for ensuring the proper implementation of AI in the creative writing process. This work demonstrates how deep learning arts and creativity may be fused together to co-create new instruments for performing poets and writers globally

REFERENCES

[1] Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, 3104-3112.

[2] Zhang, X., Lapata, M. (2014). Chinese Poetry Generation with Recurrent Neural Networks. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 670-680.

[3] Ghazvininejad, M., Shi, X., Priyadarshi, J., Knight, K. (2016). Generating Topical Poetry. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 1183- 1191.

[4] Al-Gaphari, G., Mahdy, Y. B., Shehab, M. (2018). Arabic Text Classification Using Word Embedding and Deep Learning. Jour- nal of King Saud University-Computer and Information Sciences.

[5] Farghaly, A., Shaalan, K. (2009). Arabic Natural Language Processing: Challenges and Solutions. ACM Transactions on Asian Language Information Processing (TALIP), 8(4), 1-22.

[6] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Mem- ory. Neural Computation, 9(8), 1735-1780.

[7] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Process- ing Systems, 3111-3119.

[8] Graves, A. (2012). Sequence Transduction with Recurrent Neural Networks. arXiv preprint arXiv:1211.3711.

[9] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

[10] Chen, M. X., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I. (2020). Generative Pretrained Transformers (GPT-3). OpenAI Report.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30.