Comparative Analysis of Recurrent Neural Network Architectures for Arabic Word Sense Disambiguation

Rakia Saidi[1] [a], Fethi Jarray[2] [b] and Mohammed Alsuhaibani[3] [c]

[1] *LIMTIC Laboratory, UTM University, Tunis, Tunisia*

[2] *Higher Institute of Computer Science of Medenine, Gabes University, Medenine, Tunisia*

[3] *Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia*

Keywords: Word Sense Disambiguation, Arabic text, Supervised Approach, Recurrent Neural Network.

Abstract: Word Sense Disambiguation (WSD) refers to the process of discovering the correct sense of an ambiguous word occurring in a given context. In this paper, we address the problem of Word Sense Disambiguation of low-resource languages such as Arabic language. We model the problem as a supervised sequence-to-sequence learning where the input is a stream of tokens and the output is the sequence of senses for the ambiguous words. We propose four recurrent neural network (RNN) architectures including Vanilla RNN, LSTM, BiLSTM and GRU. We achieve, respectively, 85.22%, 88.54%, 90.77% and 92.83% accuracy with Vanilla RNN, LSTM, BiLSTM and GRU. The obtained results demonstrate superiority of GRU based deep learning Model for WSD over the existing RNN models.

1 INTRODUCTION

Word Sense Disambiguation (WSD) is a natural language processing (NLP) subfield. It is the process of figuring out what a word means in context. Because natural language is intrinsically ambiguous, a single word may have multiple interpretations, making the process challenging. WSD is used in a variety of applications in real life, including semantic interpretation, web intelligence and semantic web, knowledge extraction, sentiment analysis. Due to the considerable semantic ambiguity, WSD is regarded as one of the most difficult challenges in natural language processing.

Various machine learning approaches have been proposed to automatically detect the intended meaning of a polysemous word. Deep neural networks (DNN) have recently demonstrated extraordinary capabilities and have revolutionized artificial intelligence for the majority of tasks. DNN outperforms classical learning approaches in the field of NLP in particular.

Our contribution in this paper is to propose four recurrent neural network architectures for Arabic word sense disambiguation. More specifically, we adapt Vanilla RNN, Long-short-term memory(LSTM), BiLSTM and Gated recurrent units (GRU) for WSD. We conduct experimental results on the publicly available Arabic WordNet (AWN) corpus [1]

The rest of the paper is organized as the following: Section 2 presents the state of the art. Our RNN-based system model for Arabic WSD is explained in Section 3. The results and discussions are presented in Section 4. We conclude this paper with a summary of our contribution, and we mention some future extensions.

2 STATE OF THE ART

Arabic WSD (AWSD) approaches can be classified into three categories: supervised methods, semi-supervised methods (Merhbene et al., 2013b) and unsupervised methods(Pinto et al., 2007). More than the intrinsic difficulty of WSD itself, in the Arabic case, we face the challenge of scarcity of resources.

The supervised methods use manually sense-annotated corpora to train for WSD. These approaches rely on many manually sense-tagged cor-

[a] https://orcid.org/0000-0003-0798-4834
[b] https://orcid.org/0000-0003-2007-2645
[c] https://orcid.org/0000-0001-6567-6413

[1] http://globalwordnet.org/resources/arabic-wordnet/awn-browser/

Saidi, R., Jarray, F. and Alsuhaibani, M.
Comparative Analysis of Recurrent Neural Network Architectures for Arabic Word Sense Disambiguation.
DOI: 10.5220/0011527600003318
In *Proceedings of the 18th International Conference on Web Information Systems and Technologies (WEBIST 2022)*, pages 272-277
ISBN: 978-989-758-613-2; ISSN: 2184-3252

pora, which is a time-consuming and labor-intensive task. We can divide the supervised methods into five subcategories: Machine Learning (ML) based approaches (Elmougy et al., 2008; El-Gamml et al., 2011; Merhbene et al., 2013a; Laatar et al., 2018; Eid et al., 2010; El-Gedawy, 2013; Alkhatlan et al., 2018; Bakhouche et al., 2015), Deep Learning (DL) based approaches(El-Razzaz et al., 2021; Saidi and Jarray, 2022; Al-Hajj and Jarrar, 2021), knowledge based approaches(Zouaghi et al., 2011; Zouaghi et al., 2012), information retrieval (IR) based approaches(Bouhriz et al., 2016; Alian et al., 2016; Abood and Tiun, 2017; Abderrahim and Abderrahim, 2018; Bounhas et al., 2011; Al-Maghasbeh and Bin Hamzah, 2015; Soudani et al., 2014), and metaheuristic based approach (Menai, 2014; Menai and Alsaeedan, 2012). In this manuscript, we are mainly interested in the DL based approach for WSD.

Most of the recent work in AWSD has been experimented with word embedding techniques such as word2vec (Mikolov et al., 2013a) or Glove (Pennington et al., 2014). Alkhatlan et al. (Alkhatlan et al., 2018) showed that word representation Skip-Gram method achieved higher accuracy of 82.17% compared to Glove with 71.73%. Laatar et al. (Laatar et al., 2018) used the skip-gram and CBOW models to analyze word representations to choose the optimal architecture to generate a better word embedding model for Arabic Word Sense Disambiguation. They used the Historical Arabic Dictionary Corpus, which they supplemented with 200 texts collected from Arabic Wiki Source. Skip-gram outperforms the CBOW by 51.52% accuracy where the CBOW represents 50.25%. El-Razzaz et al. (El-Razzaz et al., 2021) and (Al-Hajj and Jarrar, 2021) presented a WSD approach based on Arabic gloss which they introduced context-gloss benchmark. The authors (El-Razzaz et al., 2021) built two models that can efficiently conduct Arabic WSD using the Bidirectional Encoder Representation from Transformers (BERT). This model earns an F1-score of 89% when Al-Hajj and Jarrar (Al-Hajj and Jarrar, 2021) obtained 84% in terms of accuracy by fine-tuning three pretrained Arabic BERT models. Saidi and Jarray (Saidi and Jarray, 2022) combined the part of speech tagging POS and the BERT model for the AWSD.

There are many issues in the DL approaches devoted to Arabic WSD. For example, (El-Razzaz et al., 2021) presented a very small dataset with many redundant entries. Similarly, the dataset proposed by (Al-Hajj and Jarrar, 2021) is not publicly available. Therefore, in this paper, our sole focus is on the Arabic WordNet dataset (AWN).

Our contribution falls into centralized DL techniques, contrary to federated learning (Boughorbel et al., 2019). More precisely, we will study different variants of recurrent neural networks (RNNs), such as LSTM, bi-LSTM, and GRU.

3 RNN PROPOSED MODELS

Deep neural networks are an emerging approach and widely used in several domains such as computer vision, automatic processing of natural languages, transfer learning, text classification, etc. Usually, neural networks contain two or more hidden layers. The term ”deep” refers to the number of hidden layers of the network that can reach 150 neurons in this type of network.

In this paper, we propose a recurrent neural network (RNN) model for Arabic word sense disambiguation (AWSD). More precisely, the AWSD is simulated using architectures including simple or vanilla RNN, bidirectional long-short-term memory (bi-LSTM), gated recurrent unit (GRU) and long-short-term memory (LSTM). All models are coded in python software programming language and are available at GitHub upon request.

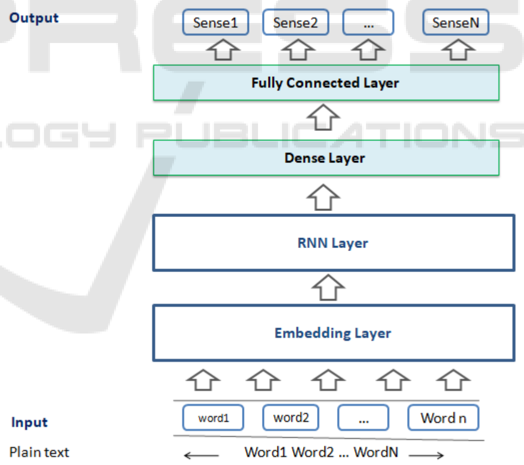

Figure 1: RNN architecture for AWSD.

Figure 1 depicts the common architecture of the different models. First, the input sentence is tokenized into tokens using Farasa (Abdelali et al., 2016). Tokenization is another challenge of Arabic texts since Arabic has a complicated morphology, and the used tool, Farasa, is handling such challenges well. Second, each token is fed into the embedding layer that produces the word embeddings and models the input sentence as a stream of vector embedding. In our implementation, we used skip-gram(Mikolov et al., 2013b) as a word embedding model. Third, the output of the embedding layer is fed into the RNN layer to get a hidden representation that takes into account the order of words in the sentence. Fourth, the hidden representations are fed into a fully dense layer and a fully-connected layer to get a more compact representation. The details of each model are shown in Figure 2.

3.1 Vanilla RNN Model for AWSD

We started by using the Vanilla RNN network. This RNN model is a sequence-to-sequence learning approach in which the input sequence represents words and the output sequence represents meaning. The studies were carried out using Arabic embedding that had been pre-trained.

3.2 Long Short-Term Memory (LSTM)

The LSTM is a recurrent extension of the neural network. This sort of neural network was created to solve the problem of long-term dependence. It is capable of retaining knowledge for a long time. This form of network can determine what to keep in the long-term state, what to delete, and what to read. The long-term state path goes through a forget gate, which removes some information, then new ones are added, which are selected by an input gate, and the result is given without further modification. The long-term state is copied and then sent to the tanh function before being filtered by the output gate after this addition operation.

3.3 Bidirectional Long Short-Term Memory (Bi-LSTM)

Bidirectional LSTM is a variant of standard LSTM that can help increase model performance when dealing with sequence classification problems. The Bi-LSTM blends two LSTMs when the entire sequence is accessible. One is based on the input sequence, while the other is based on a reverse replica of the input sequence.

3.4 Gated Recurrent Unit (GRU)

The GRU cell is a simpler variant of the LSTM cell that is becoming increasingly popular as a result of its superior performance.

4 EXPERIMENTS

We will compare different RNN architectures for Arabic word sense disambiguation. So different experiments are made while fixing each time a hyperparameter. As a performance metric, we used accuracy as a percentage of correctly predicted word senses.

4.1 Dataset

In this paper, we use the publicly available AWN [2]. AWN is a lexical resource for modern standard Arabic. It was constructed according to the linguistic resources of Princeton WordNet. It is structured around elements called Synsets, which are a list of synonyms and pointers linking it to other synsets. Currently, this corpus has 23.481 words grouped into 11.269 synsets. One or more synsets may contain a common word. The senses of a word are related by the synsets in the AWN to which it belongs. We suppose that the dataset set is clean, as opposite to noisy dataset (Boughorbel et al., 2018)

4.2 Hyperparameters Setting

We have several parameters to set for the model. Standard parameters are the word embedding size, the number of epochs during training, and the batch size. In the testing process, we randomly select 80% of the dataset as a training set, 10% as a test set and the remaining 10% as a validation set. In all conducted experiments, the ratio remains the same. And for the record, all the results are obtained on the test set. run while changing the size of the hyperparameters (batch size and epochs number).

The batch size is the number of training samples for mini-batches. It is one of the most critical hyperparameters to tune in neural network learning. Table1 presents the effects of batch size on WSD performance. A smaller batch size necessitates more calculations and model weight updates, which takes longer. Larger batch sizes require less effort and are faster to execute. However, according to Table1, the large and small batch sizes lead to poor generalization. In the following, we set the batch size to 64.

Table 2 displays the effect of the number of epochs on the generalization ability of different RNN architectures. We note that generalization and the number of epochs are not monotonically related. For a few epochs, such as 100, we get low accuracy because the models may underfit the data. Similarly, when we increase the number of epochs too much, the accuracy

[2] http://globalwordnet.org/resources/arabic-wordnet/awn-browser/

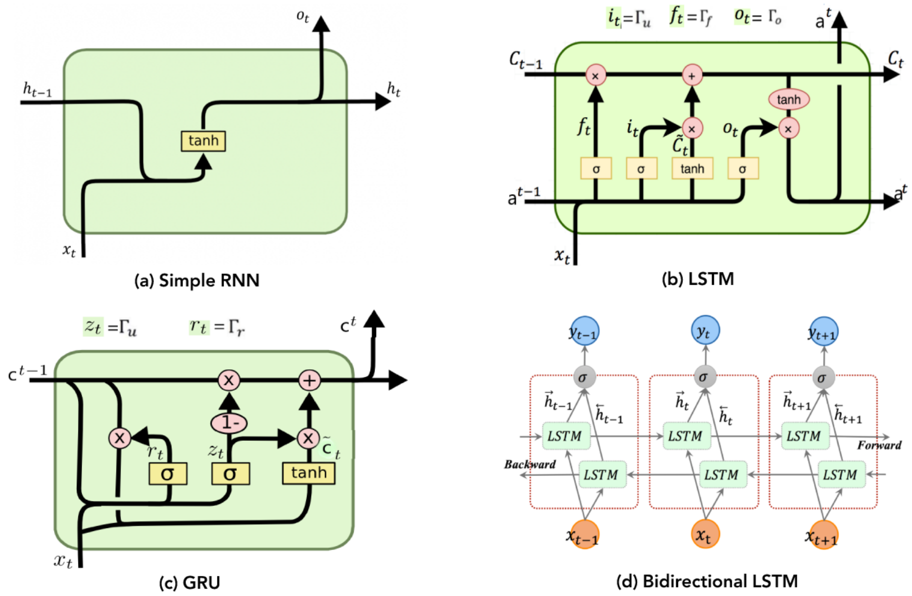

Figure 2: The general structure of recurrent neural network models. (a) Simple RNN, (b) LSTM, (c) GRU, and (d) Bi-LSTM models (Apaydin et al., 2020)

Table 1: Effect of batch size. Number of epochs to 50.

Method	batch-size	Predicting accuracy
Vanilla RNN	28	77.3%
	32	80.5%
	64	**83.08%**
	128	81.2%
LSTM	28	81.26%
	32	83.5%
	64	**86.87%**
	128	84.2%
BiLSTM	28	82.78%
	32	84.4%
	64	**88.95%**
	128	86.16%
GRU	28	83.9%
	32	86.5%
	64	**91.02%**
	128	87.94%

Table 2: Effect of number of epochs. Batch size=64.

Method	Epochs	Predicting accuracy
Vanilla RNN	50	83.08%
	100	**85.22%**
	150	85.22%
	200	85.22%
LSTM	50	86.87%
	100	**88.54%**
	150	88.54%
	200	88.54%
BiLSTM	50	88.95%
	100	**90.77%**
	150	90.15%
	200	90.15%
GRU	50	91.02%
	100	**92.83%**
	150	92.83%
	200	92.83%

does not increase because the models could overfit the training data. In the following, we fix the number of epochs to 100.

5 RESULTS AND DISCUSSION

According to Tables 1 and 2, we fixed the batch size at 64 and the number of epochs to 100. Table 3 summarizes the results obtained with a Skip-Gram word embedding of dimension 300 and Adam for optimizer and compares it with the most recent methods based on AWN dataset.

For our approach and to compare the RNN models for ASWD, Table 3 shows that the best is the GRU-based model, which beats even the LSTM-based model. This could be because GRU utilizes less memory and trains faster than LSTMs and BiLSTM models because it reduces the number of gates and has fewer training parameters. The obtained results illustrate the effectiveness of the proposed GRU based model.

Finally, it is worth noting the difficulty of making a fair comparison between DL models devoted to Ara-

Table 3: Accuracy of RNN models for AWSD.

Model	Accuracy
GRU	**92.83%**
BiLSTM	90.77 %
LSTM	88.54%
Vanilla RNN	85.22 %
Glove (Alkhatlan et al., 2018)	71.73%
Skip-Gram(Alkhatlan et al., 2018)	82.17%
IR (Bouhriz et al., 2016)	74%

bic language because different datasets are employed. For example, (El-Razzaz et al., 2021) and (Al-Hajj and Jarrar, 2021), as already noted, used a Gloss-type dataset. This variety in models and datasets encourages us to combine our GRU method with the BERT model in a future contribution (Chouikhi et al., 2021).

6 CONCLUSION

In this paper, we propose a recurrent neural network-based architecture for Arabic text to solve the problem of word-sense disambiguation. We validate our approach through the Arabic WordNet dataset. We show that that GRU model outperforms the other RNN models and achieves about a 93 percent of prediction accuracy. As a future work, we plan to use the more advanced word embedding such as BERT for the embedding step and combine it with the RNN models.

REFERENCES

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations*, pages 11–16.

Abderrahim, M. A. and Abderrahim, M. E. A. (2018). Arabic word sense disambiguation with conceptual density for information retrieval. *Models and Optimisation and Mathematical Analysis Journal*, 6(1):5–9.

Abood, R. H. and Tiun, S. (2017). A comparative study of open-domain and specific-domain word sense disambiguation based on quranic information retrieval. In *MATEC Web of Conferences*, volume 135, page 00071. EDP Sciences.

Al-Hajj, M. and Jarrar, M. (2021). Arabglossbert: Fine-tuning bert on context-gloss pairs for wsd.

Al-Maghasbeh, M. K. A. and Bin Hamzah, M. (2015). Extract the semantic meaning of prepositions at arabic texts: an exploratory study. *Int J Comput Trends Technol*, 30(3):116–120.

Alian, M., Awajan, A., and Al-Kouz, A. (2016). Word sense disambiguation for arabic text using wikipedia and vector space model. *International Journal of Speech Technology*, 19(4):857–867.

Alkhatlan, A., Kalita, J., and Alhaddad, A. (2018). Word sense disambiguation for arabic exploiting arabic wordnet and word embedding. *Procedia computer science*, 142:50–60.

Apaydin, H., Feizi, H., Sattari, M. T., Colak, M. S., Shamshirband, S., and Chau, K.-W. (2020). Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. *Water*, 12(5):1500.

Bakhouche, A., Yamina, T., Schwab, D., and Tchechmedjiev, A. (2015). Ant colony algorithm for arabic word sense disambiguation through english lexical information. *International Journal of Metadata, Semantics and Ontologies*, 10(3):202–211.

Boughorbel, S., Jarray, F., Venugopal, N., and Elhadi, H. (2018). Alternating loss correction for preterm-birth prediction from ehr data with noisy labels. *arXiv preprint arXiv:1811.09782*.

Boughorbel, S., Jarray, F., Venugopal, N., Moosa, S., Elhadi, H., and Makhlouf, M. (2019). Federated uncertainty-aware learning for distributed hospital ehr data. *arXiv preprint arXiv:1910.12191*.

Bouhriz, N., Benabbou, F., and Lahmar, E. B. (2016). Word sense disambiguation approach for arabic text. *International Journal of Advanced Computer Science and Applications*, 7(4):381–385.

Bounhas, I., Elayeb, B., Evrard, F., and Slimani, Y. (2011). Organizing contextual knowledge for arabic text disambiguation and terminology extraction. *KO KNOWLEDGE ORGANIZATION*, 38(6):473–490.

Chouikhi, H., Chniter, H., and Jarray, F. (2021). Arabic sentiment analysis using bert model. In *International Conference on Computational Collective Intelligence*, pages 621–632. Springer.

Eid, M. S., Al-Said, A. B., Wanas, N. M., Rashwan, M. A., and Hegazy, N. H. (2010). Comparative study of rocchio classifier applied to supervised wsd using arabic lexical samples. In *Proceedings of the tenth conference of language engeneering (SEOLEC'2010), Cairo, Egypt*.

El-Gamml, M. M., Fakhr, M. W., Rashwan, M. A., and Al-Said, A. B. (2011). A comparative study for arabic word sense disambiguation using document preprocessing and machine learning techniques. *ALTIC , Alexandria, Egypt*.

El-Gedawy, M. N. (2013). Using fuzzifiers to solve word sense ambiguation in arabic language. *International Journal of Computer Applications*, 79(2).

El-Razzaz, M., Fakhr, M. W., and Maghraby, F. A. (2021). Arabic gloss wsd using bert. *Applied Sciences*, 11(6):2567.

Elmougy, S., Taher, H., and Noaman, H. (2008). Naïve bayes classifier for arabic word sense disambiguation. In *proceeding of the 6th International Conference on Informatics and Systems*, pages 16–21. Citeseer.

Laatar, R., Aloulou, C., and Belghuith, L. H. (2018). Word2vec for arabic word sense disambiguation. In *International Conference on Applications of Natural*

Language to Information Systems, pages 308–311. Springer.

Menai, M. E. B. (2014). Word sense disambiguation using evolutionary algorithms–application to arabic language. *Computers in Human Behavior*, 41:92–103.

Menai, M. E. B. and Alsaeedan, W. (2012). Genetic algorithm for arabic word sense disambiguation. In *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 195–200. IEEE.

Merhbene, L., Zouaghi, A., and Zrigui, M. (2013a). An experimental study for some supervised lexical disambiguation methods of arabic language. In *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*, pages 1–6. IEEE.

Merhbene, L., Zouaghi, A., and Zrigui, M. (2013b). A semi-supervised method for arabic word sense disambiguation using a weighted directed graph. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1027–1031.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013b). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pinto, D., Rosso, P., Benajiba, Y., Ahachad, A., and Jiménez-Salazar, H. (2007). Word sense induction in the arabic language: A self-term expansion based approach. In *Proc. 7th Conference on Language Engineering of the Egyptian Society of Language Engineering-ESOLE*, pages 235–245. Citeseer.

Saidi, R. and Jarray, F. (2022). Combining bert representation and pos tagger for arabic word sense disambiguation. In *International Conference on Intelligent Systems Design and Applications*, pages 676–685. Springer.

Soudani, N., Bounhas, I., ElAyeb, B., and Slimani, Y. (2014). Toward an arabic ontology for arabic word sense disambiguation based on normalized dictionaries. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 655–658. Springer.

Zouaghi, A., Merhbene, L., and Zrigui, M. (2011). Word sense disambiguation for arabic language using the variants of the lesk algorithm. *WORLDCOMP*, 11:561–567.

Zouaghi, A., Merhbene, L., and Zrigui, M. (2012). A hybrid approach for arabic word sense disambiguation. *International Journal of Computer Processing Of Languages*, 24(02):133–151.