Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Arabic Computational Linguistics (ACLing 2018), November 17-19 2018, Dubai, United Arab Emirates

Implementation of A Neural Natural Language Understanding Component for Arabic Dialogue Systems

Abdallah M. Bashir[a], Abubakr Hassan[a], Benjamin Rosman[b], Daniel Duma[c], Mohanad Ahmed[a]

[a] *University of Khartoum, Khartoum, Sudan*
[b] *University of the Witwatersrand / CSIR, South Africa*
[c] *University of Edinburgh , United Kindom*

Abstract

Natural Language Understanding (NLU) is considered a core component in implementing dialogue systems. NLU has been greatly enhanced by deep learning techniques such as word embeddings and deep neural network architectures, but current NLP methods for Arabic language dialogue action classification or semantic decoding is mostly based on handcrafted rule-based systems and methods that use feature engineering, but without the benefit of any form of distributed representation of words. This paper presents an approach to use deep learning techniques for text classification and Named Entity Recognition for the domain of home automation in Arabic. To this end, we present an NLU module that can further be integrated with Automatic Speech Recognition (ASR), a Dialogue Manager (DM) and a Natural Language Generator (NLG) module to build a fully working dialogue system. The paper further describes our process of collecting and annotating the data, structuring the intent classifier and entity extractor models, and finally the evaluation of these methods on different benchmarks.

Keywords: Natural language understanding; Semantic Decoding; NLP; Arabic; Dialogue Systems; Named Entity Recognition; Text Classification

[1] Both authors contributed equally in this work
[*] Abdallah M. Bashir. Tel.: +249-900-776-902
E-mail address: abdullahbashir077@gmail.com

1. Introduction

Technological advances aim to ease the interaction between users and computer systems and speaking naturally with the computer is one such form. Although deep learning has revolutionized the natural language processing area for high-resource languages like English, Arabic is lagging behind in this revolution.

Dialogue systems or conversational agents represent one of the most important applications of Natural Language Processing, and in particular task-oriented Dialogue Systems as these become a viable solution to automating tasks such as restaurant reservations and booking airline tickets. Another popular use of task-oriented dialogue systems is Home Automation. For example, as of mid-2018[2] Google have reportedly shipped 3.2 million of its Google Home and Home Mini devices which, in addition to the 2.5 million Echo devices shipped by Amazon in the first quarter of 2018, indicates the popularity of Home Automation assistants.

According to Chen et al. [1] the structure of Task-oriented Dialogue Systems requires a number of components including Natural Language Understanding or Semantic Decoding, Dialogue State Tracking, Dialogue Policy Learning and Natural Language Generation. The Natural Language Understanding unit represents the main component to understand the user's input to the dialogue system as it classifies the users intent and extracts the target and desired settings of this intent.

By reviewing the literature on Arabic language, it appears that Arabic has not received much attention from the recent state-of-the-art approaches to Natural Language Understanding. Most methods use rule-based systems which can prove effective for small tasks but cannot scale to real world applications which are complex by nature, and have a high cost of maintenance [2].

In this paper we present a neural network implementation of a Natural Language Understanding component for Arabic task-oriented Dialogue Systems; the task that the system will be based on in this case is home automation. This module is composed of an Intent Classifier and a Slot Tagger that work together as an understanding component to decode input commands to the Dialogue System.

The structure of this paper is as follows. The process of collecting and annotating the data is discussed in section 3.1. We then describe the system architecture, starting with the Intent Classifier in section 3.2 and the Entity Extractor in section 3.3. In these sections we demonstrate the potential of using deep learning techniques in text classification and Named Entity Recognition. Finally, in section 4 we present some evaluation measures for each component.

2. Related Work

There are numerous contributions targeting the English language regarding Neural implementations for Dialogue Systems. Vinyals et al. [4] presented a general purpose conversational agent that can generate simple and basic conversations, and extract knowledge from a noisy but open-domain dataset by training an end-to-end model. There are also a number of implementations when it comes to task-oriented Dialogue Systems for tasks such as restaurants search, airline tickets reservation and home automation. Wen et al. [5] presented a novel neural network-based framework for task-oriented Dialogue Systems in a restaurant search domain.

For text classification many techniques have been used, such as Convolutional Neural Networks (CNN). Kim [6] shows that a CNN can achieve high results on multiple benchmarks and are considered faster than other techniques such as sequential models which are more commonly applied to these kinds of problems. Some methods combine both approaches, such as Lee et al. [7].

For Named Entity Recognition Chiu et al. [8] use CNNs to extract features from character embeddings of each word and then feed these features into a Long-Short Term Memory (LSTM) recurrent neural network along with the vector representation of that word. Other techniques benefit from the relation between sequence labels to enhance tagging using Conditional Random Fields (CRF) [9] along with CNNs and LSTMs.

Despite all these solutions, a task-oriented Dialogue System has not yet been explored for the Arabic language using neural networks. In a similar space however, Moubaiddin et al. [10] implemented an Arabic Dialogue System

[2] https://www.forbes.com/sites/johnkoetsier/2018/05/25/massive-reversal-google-home-sales-explode-483-to-beat-amazon-for-smart-speaker-crown/

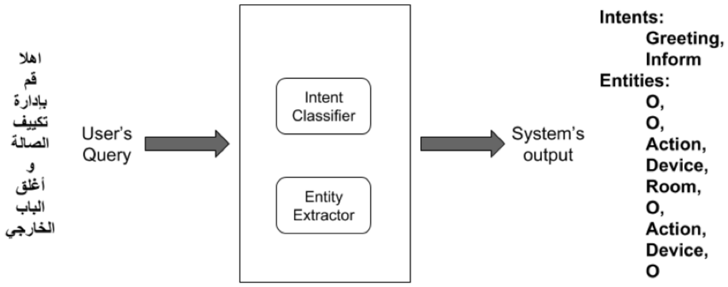

Fig. 1: Systems components

intended to interact with hotel customers and generate responses about reserving a hotel room and other services. However, their approach was to use a grammatical parser that is based on Government Binding Theory [11]. In another example, a rule-based approach for building a conversational agent was presented by Botta in Abu Ali et al. [12], who used AIML (Artificial Intelligence Markup Language) to implement a chatbot which was based on the Egyptian Arabic dialect. Elmadani et al. [13] demonstrated a similar approach to the one proposed in this paper by presenting an effective dialogue actions classification model using a Support Vector Machine (SVM) as a classification technique.

3. Method

We implement the language understanding module using two components: The Intent Classifier and the Entity Extractor. The Intent Classifier was implemented using two neural text classification techniques, which are LSTMs and CNNs. The Entity Extractor was implemented using a Bidirectional LSTM along with character-based word embeddings. All models were optimized using Dropout and Early Stopping techniques. Fig. 1. demonstrates the system components with an example input query ("Hello, turn on the halls air conditioner and close the outer door") the system will recognize the intents embedded in the sentence as (Greeting, Inform) and will tag each word with a label that represents its role in the sentence, O means that the word represents none of the predefined tags.

3.1. Dataset

3.1.1. Data Collection and annotation

Statistical approaches to Dialogue Systems require considerable time and effort when it comes to collecting data, especially in the case of task-oriented Dialogue Systems since the data needs to be in-domain and carefully labelled. The main domain of the dialogue application in our case, as mentioned above, is in Home Automation. To collect the data for this application we used an online survey as there were no available sources for previous labeled datasets in Arabic oriented to Home Automation. After filtering and labelling the collected data, the result was 768 entries. The data was labeled using two approaches: the first was to classify the intents in each sentence (this was done directly in the survey), and in the second approach we labeled sentences according to the Conll-2003 NER format (Sang et al. [14]). We also used the AQMAR dataset [15], which is a NER corpus extracted from Wikipedia text, and we took all the PER tags which are examples of people names to extract it along with our entities for a more flexible dialogue experience.

3.1.2. Data Pre-processing

It is common for Arabic text to have many letters which can be mistaken for others as a result of their visual similarity, so a common practice among Arabic linguistics is to normalize text, i.e. to group all groups of miswritten letters together. This may introduce some ambiguity but reduces the complexity of the Arabic text [16].

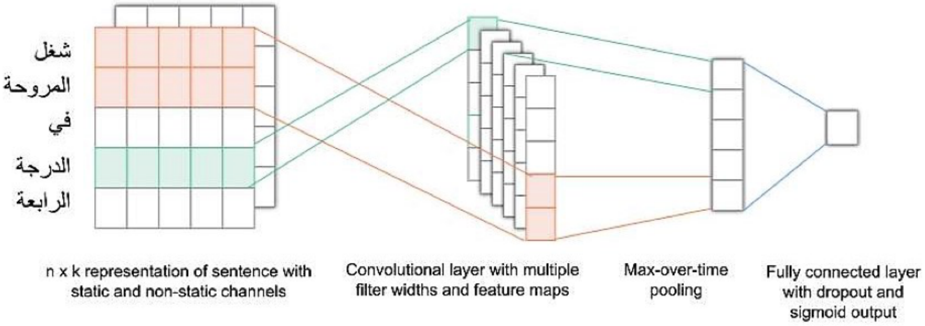
شغل
المروحة
في
الدرجة
الرابعة
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and sigmoid output

Fig. 2: Intent classification using CNN Inspired from Kim [6]

3.1.3. *Data Representation*

Our approach to modeling is based on neural networks which only work with numeric values. Vectorizing text is required to transform text to numeric tensors. Text Vectorization can be based on characters, words or n-grams of words. One form of Text Vectorization is Word Vectors [3] which are a distributional representation of a word's meaning. Word vectors imply potential relationships, such as contextual closeness, which are captured across collections of words. The usage of Word Vectors here is beneficial as they accelerate the process of collecting data as the models require less data to understand a large number of words. We use AraVec [17], a pre-trained word embedding model for Arabic to represent text data in our models.

3.2. *Intent Classifier*

This component is responsible for classifying the users intent in order to direct the Dialogue System to the appropriate answer. The data was constructed as a set of sentences each labeled with the intents it contains. The entries in our collected data was labeled with five intents: *greeting, inform, check_status, chat*, and *goodbye*.

We applied two neural approaches to implement the intent classifier, namely CNNs and LSTMs, to demonstrate the potential of these methods in identifying the users intent. Here we describe each model in detail, while the results are discussed in section 4 with some evaluation measurements.

For both models, input is converted from a sequence of text to a sequence of word embeddings before training. Word embeddings are fed to the model one at a time and encoded (an internal representation with features that the models have seen useful for prediction) at the final output point of the model. This approach is known as encoding, and the output of the encoder is passed through a fully connected sigmoid layer, which computes the probability of the presence of each intent individually.

3.2.1. *CNNs*

Kims [6] work on applying convolutional neural networks in sentence classification suggested that a simple CNN with one layer of convolution performs remarkably well on this task. Inspired by this we implemented a single layer of convolution to the sentences matrix, where each column represents a sentence as a group of words, and each row represents the word vector for the current word. Fig. 2. illustrates the general structure of the model with an input example of (Start the fan at the fourth speed).

$$\begin{aligned} z &= CNN(x_1, x_2, x_3, ..., x_n) \\ y &= \sigma(z) \end{aligned} \tag{1}$$

3.2.2. *LSTMs*

LSTM, or Long Short-Term Memory, as presented by Hochreiter, et al. [18], is considered one of the most powerful techniques in NLP for the ability to capture long-term relations between parts of some text. This approach has proved

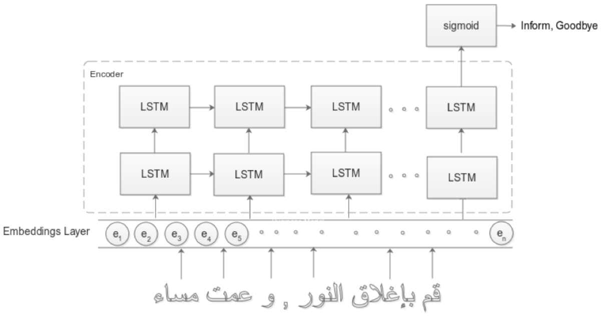

Fig. 3: Intent classification using LSTM

to be among the most accurate for text classification, as will be shown in section 4. Fig. 3. illustrates the general structure of the model with an input example (Turn off the light and good night), each cell represents a timestep over the input sequence.

$$
\begin{aligned}
z &= LSTM(x_1, x_2, x_3, ..., x_n)\\
y &= \sigma(z)
\end{aligned}
\tag{2}
$$

It should be noted that the number of time steps is fixed such that zero padding is applied, where each sentence is padded with vectors of zeros until it reaches the input word count.

3.3. Entity Extractor

This component extracts the main tags from commands. It works by giving each word in the sentence a label that identifies its role. For our system we considered five entities to be tagged, namely: Room, Device, Actions applied to devices, Device Speeds and Person names.

To label entities we added a new representation of the input words which is character generated word embeddings, these are considered a useful way to handle Out-Of-Vocabulary words (OOV) [19]. Here word embeddings are generated from the sequence of characters that construct the word using a Bidirectional LSTM. Fig.4 shows the Arabic words (open) characters propagate the model in both forward and backward directions, the characters get encoded in each cell (diamond shape) until the last cell in each layer, the output of both layers is combined to produce the final word vector shape The output of the Bidirectional LSTM (BiLSTM) is combined with the vector representation of the word (fetched from the pre-trained word embeddings). If the word is not found in the vocabulary, then a special token <UNK> is assigned to it. This combined vector is passed to a BiLSTM (See Fig.5) that encodes the sentence at each

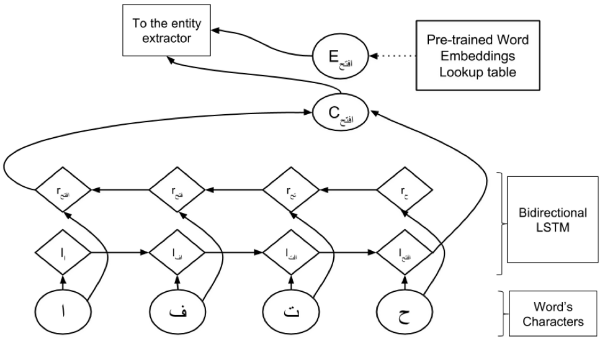

Fig. 4: Generating character based word embeddings

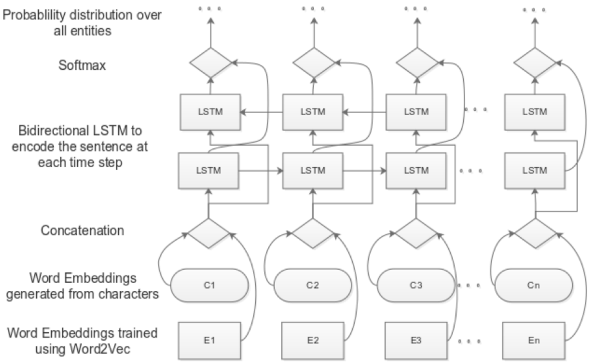
Probablility distribution over all entities
Softmax
Bidirectional LSTM to encode the sentence at each time step
Concatenation
Word Embeddings generated from characters
Word Embeddings trained using Word2Vec
LSTM
C1
C2
C3
Cn
E1
E2
E3
En

Fig. 5: BiLSTM model for Named Entity Recognition

time step, and this encoding is passed to a SoftMax layer that computes a probability distribution over all the entities. This approach is motivated by Lample et al. [20]. Fig. 4 and Fig. 5 illustrate this method visually.

4. Evaluation

The quantitative metrics of F-Score, precision, recall, and accuracy were used to evaluate our Intent Classifier and Entity Extractor models, for both components, the data was divided to train and test sets with 70%, 30% respectively.

We ran number of experiments to determine the best number of layers regarding using the CNN and the LSTM in text classification for our case, table. 1 shows that the option of using one layer for the CNN and two layers for the LSTM achieved the highest results for both cases.

Table 1: Intent classification with CNN and LSTM: number of layers optimization

Architecture + Layers	Accuracy	F-Score	Precision	Recall
CNN 1 layer	**97.29**	**94.17**	**96.4**	92.14
CNN 2 layers	78.24	84.85	96.86	75.65
LSTM 1 layer	95.3	90.07	90.6	89.62
LSTM 2 layers	**96.33**	**92.25**	**92.29**	**92.22**

Table. 2 shows that the CNN performance is generally better than the LSTM when applied to the dataset mentioned earlier. This shows the potential of such methods for text classification in Arabic.

Table 2: CNN vs RNN for Intent Classification tests results.

Model Type	Accuracy	F-Score	Precision	Recall
CNN	**97.29**	**94.17**	**96.4**	92.14
LSTM	96.33	92.25	92.29	**92.22**

Table. 3 shows the results of the Entity Extraction model which includes char embeddings fed to a BiLSTM network. Usage of the BiLSTM allows for capturing the patterns that each entity occurs in, and the character generated word embeddings enhance the model by capturing the meaning of Out-Of-Vocabulary (OOV) words. Table. 3 shows that using Char Embeddings results in a high increase in the measures. Comparing to other results shown in Elmadany et, al. [13] our approach's result surpasses all of them.

Table 3: Entity Extraction Evaluation tests results

Model Type	Accuracy	F-Score	Precision	Recall
BiLSTM	95.05	90.00	88.00	91.00
BiLSTM + Char Embeddings	**97.81**	**94.00**	**92.00**	**95.00**

5. Conclusion and Future Works

In this paper we presented a neural network approach to implementing a Natural Language Understanding unit for Arabic task-oriented Dialogue Systems. Towards our task of home automation, we collected and labeled a dataset to be used for training and testing the models. We used state-of-the-art neural network text classification techniques of CNN and LSTM to classify the users intent. Both of the intent classification implementations were benchmarked and the results of the benchmark indicated that the LSTM performance with an F-Score of 92.01 was slightly better than the CNNs performance. For extracting the user targets and goals from the input, we used a combined representation of word embeddings and character based word embeddings which are fed later to a Bidirectional LSTM network. The BiLSTM with the Char Embeddings model achieved a high F-Score of 94.0 which implies that its performance is very similar to current Named Entity Recognition benchmarks in English. In future work, our Natural Language Understanding module can be integrated with Automatic Speech Recognition (ASR) and Natural Language Generation (NLG) modules to yield an efficient task oriented Dialogue System.

References

[1] Chen H, Liu X, Yin D, Tang J. A Survey on Dialogue Systems: Recent Advances and New Frontiers. arXiv:171101731 [cs] [Internet]. 2017 Nov 6

[2] MrkÅi N. Data-Driven Language Understanding for Spoken Dialogue Systems [Internet] [Thesis]. University of Cambridge; 2018. p 13 Available from: https://www.repository.cam.ac.uk/handle/1810/276689

[3] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed Representations of Words and Phrases and their Compositionality. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Advances in Neural Information Processing Systems 26 [Internet]. Curran Associates, Inc.; 2013 . p. 31113119. .

[4] Vinyals O, Le Q. A neural conversational model. arXiv preprint arXiv:150605869. 2015;

[5] Wen T-H, Vandyke D, Mrksic N, Gasic M, Rojas-Barahona LM, Su P-H, et al. A Network-based End-to-End Trainable Task-oriented Dialogue System. arXiv:160404562 [cs, stat] [Internet]. 2016 Apr 15

[6] Kim Y. Convolutional Neural Networks for Sentence Classification. arXiv:14085882 [cs] [Internet]. 2014 Aug 25

[7] Lee JY, Dernoncourt F. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. arXiv:160303827 [cs, stat] [Internet]. 2016 Mar 11

[8] Chiu JPC, Nichols E. Named Entity Recognition with Bidirectional LSTM-CNNs. arXiv:151108308 [cs] [Internet]. 2015 Nov 26

[9] Ma X, Hovy E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. arXiv:160301354 [cs, stat] [Internet]. 2016 Mar 4

[10] Moubaiddin A, Shalbak O, Hammo B, Obeid N. Arabic Dialogue System for Hotel Reservation based on Natural Language Processing Techniques. Computacin y Sistemas. 2015 Mar 27;19:16.

[11] Black, C. (1999). A step-by-step introduction to the government and binding theory of syntax. SIL - Mexico Branch and University of North Dakota

[12] Abu Ali D, Habash N. Botta: An Arabic Dialect Chatbot. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations [Internet]. Osaka, Japan: The COLING 2016 Organizing Committee; 2016 p. 208212.

[13] Elmadany A, M Abdou S, Gheith M. Improving Dialogue Act Classification for Spontaneous Arabic Speech and Instant Messages at Utterance Level. In 2018.

[14] Sang EFTK, De Meulder F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. arXiv:cs/0306050 [Internet]. 2003 Jun 12

[15] Mohit B, Schneider N, Bhowmick R, Oflazer K, Smith NA. Recall-Oriented Learning of Named Entities in Arabic Wikipedia. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics [Internet]. Avignon, France: Association for Computational Linguistics; 2012 . p. 162 173.

[16] Farghaly A, Farghaly A, Shaalan K, Khaled. Arabic Natural Language Processing: Challenges and Solutions. ACM Transactions on Asian Language Information Processing (TALIP). 2009 Jan 1;8:1.

[17] Mohammad AB, Eissa K, El-Beltagy S. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. Procedia Computer Science. 2017 Nov 5;117:25665..

[18] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):17351780.

[19] Ling W, Lus T, Marujo L, Astudillo RF, Amir S, Dyer C, et al. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. arXiv:150802096 [cs] [Internet]. 2015 Aug 9 .

[20] Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural Architectures for Named Entity Recognition. arXiv:160301360 [cs] [Internet]. 2016 Mar 4