Empirical Evaluation of Word Representations on Arabic Sentiment Analysis

Chapter *in* Communications in Computer and Information Science · January 2018

DOI: 10.1007/978-3-319-73500-9_11

3 authors, including:

Hatem Haddad

Manouba University

Hala Mulki

Selçuk University

Empirical Evaluation of Word Representations on Arabic Sentiment Analysis

Mourad Gridach[1], Hatem Haddad[2], and Hala Mulki[3]

[1] High Institute of Technology
Ibn Zohr University, Agadir, Morocco
m.gridach@uiz.ac.ma
[2] Department of Computer and Decision Engineering
Université Libre de Bruxelles,Belgium
Hatem.Haddad@ulb.ac.be
[3] Department of Computer Engineering
Selcuk University, Konya, Turkey
halamulki@selcuk.edu.tr

Abstract Sentiment analysis is the Natural Language Processing (NLP) task that aims to classify text to different classes such as positive, negative or neutral. In this paper, we focus on sentiment analysis for Arabic language. Most of the previous works use machine learning techniques combined with hand engineering features to do Arabic sentiment analysis (ASA). More recently, Deep Neural Networks (DNNs) were widely used for this task especially for English languages. In this work, we developed a system called CNN-ASAWR where we investigate the use of Convolutional Neural Networks (CNNs) for ASA on 2 datasets: ASTD and SemEval 2017 datasets. We explore the importance of various unsupervised word representations learned from unannotated corpora. Experimental results showed that we were able to outperform the previous state-of-the-art systems on the datasets without using any kind of hand engineering features.

Keywords: Arabic language, Arabic Sentiment Analysis, Convolutional Neural Networks, Pretrained Word Representations

1 Introduction

Sentiment analysis [1], also known as opinion mining [2], is a Natural Language Processing (NLP) task that receives much attention these years where the main goal is to classify text to sentiment classes such as positive or negative, or more fine-grained classes such as very positive, positive, neutral, etc. In the last years, sentiment analysis plays an essential role where it helps to develop many online applications for customer reviews and public opinion analysis [2,3,4,5]. The classical sentiment analysis systems focus on classical opinion such as binary classification (positive or negative), while others developed systems for multiple categories such as six basic emotions (anger, happiness, fear, sadness, disgust, and surprise) [6]. Sentiment systems can then be used to identify sentiment categories from texts.

Most of the previous work on sentiment analysis developed systems for English language because resources are publicly available for the community. For Arabic language, there has been less progress, but in the last few years, more resources are freely available for the Arabic language community especially after integrating the Arabic sentiment classification as one of the shared tasks in SemEval workshop in 2016. Most of the previous work in ASA used hand engineering features [7] or machine learning approaches such as SVM combined with features such as morphological features, POS tagging, etc. [8,9]. More recently, [10] explored different deep learning architectures to do Arabic sentiment classification.

In recent years, deep neural networks are widely used machine learning models. They have shown large success by achieving state-of-the-art results in various NLP applications such as dependency parsing [11], language modeling [12], question answering [13,14], speech recognition [15] and machine translation [16,17]. In addition, deep neural networks are widely used for sequence modeling tasks such as Named Entities Recognition (NER), Part-of-Speech (POS) tagging, recurrent neural networks (RNN) and obtained state-of-the-art in various languages such as English [18], German [19], Italian [20] and Arabic [21]. Various RNN models, like Long-Short Term Memory (LSTM) [22,23] and Gated Recurrent Unit (GRU) [16], have shown success in modeling sequential data like speech recognition [15] and POS tagging [24].

In this paper, we present CNN-ASAWR (Convolutional Neural Networks for Arabic Sentiment Analysis using Word Representations), our system uses Convolutional Neural Networks (CNN) as a deep learning model for ASA. Originally invented by [25] for pattern recognition and rediscovered after that by [26], convolutional neural networks are widely used mainly in various applications in computer vision where they obtained the state-of-the-art results, as well as natural language processing. In image classification, CNN based models (AlexNet, ZF Net, GoogLeNet, VGGNet and ResNet) won the ImageNet competition since 2012 [55]. Other computer vision applications using CNN are object detection [27,28], image segmentation [29], image captioning [30,31] and more recently, they are used in self driving cars [14,32].

In NLP, CNN received less attention from the NLP community compared to computer vision. More recently, CNNs are used in many NLP applications and have been shown to be effective in NER [33,18], sentence modelling [34], search query retrieval [35], semantic parsing [36] and more recently in machine translation [37] and text summarization [38].

In this paper, we present a deep learning model based on convolutional neural network for ASA. We explored the use of different word representations trained on unannotated corpora. We investigate three main word representations: Stanford Glove vectors, Skip-gram (SG) model and Continuous bag of words (CBOW) model. These Arabic word embeddings developed by [39] are publicly available for the community. The final architecture of our model can be described as a

CNN trained on top of word representations. Despite little tuning of hyperparameters, our model achieves excellent results on two datasets, suggesting that the pretrained word representations are universal feature extractors that can be used for various classification tasks.

Experiments on SemEval 2017 and ASTD dataset that are the two largest available datasets for the community showed that we were able to get the state-of-the-art results by outperforming the previous best system on ASTD dataset that uses SVM classifier with features by a large margin. In addition, CNN-ASAWR outperforms the winner system of the SemEval 2017 shared task on ASA.

The rest of this paper is structured as follows. In section 2, we discuss the related work done in Arabic sentiment analysis and in section 3 we present our ASA approach which is based on CNN trained on pretrained word representations. The experimental results will be presented in section 4. Finally, we present the conclusion with the future work in section 5.

2 Related Work

In this section, we present an overview of the most dominant approaches in ASA. In general, most of the previous Arabic sentiment analysis systems used machine learning approaches where some systems used supervised methods while others used unsupervised methods. Furthermore, these systems combined the previous machine learning approaches with features to perform the Arabic sentiment classification task.

[40] built a system where they focused on conducting sentiment classification at document level. The authors developed a method called EWGA (Entropy Weighted Genetic Algorithm), which is a hybridized genetic algorithm that incorporates the information-gain heuristic for feature selection. This method allowed the authors to improve performance and get a better assessment of key features. They evaluate this method on a benchmark movie review dataset and U.S. and Middle Eastern Web forum postings. The experimental results using EWGA with SVM indicate high performance levels, with accuracies of over 91% on the benchmark dataset as well as the U.S. and Middle Eastern forums.

[56] performed sentence-level sentiment classification for MSA. After a series of experiments, the authors discovered that the appearance of a positive or negative adjective, based on their lexicon, is the most influential feature. [7] developed a sentiment analysis system for both Modern Standard Arabic (MSA) news articles and dialectal Arabic microblogs from Twitter. Their model used many features such as stemming, part-of-speech tagging and tweet specific features. Finally, they extend the Arabic lexicon using Arabic-English phrase tables by adopting a random graph walk approach.

[8] developed a system called SAMAR for ASA based on SVM to do classification. This system uses lots of features such as morphological features: word forms and POS tagging, standard features such as UNIQUE (Q) feature and Po-

larity Lexicon (PL), dialectal Arabic features and lastly genre specific features (Gender, User ID, and Document ID).

[9] developed an ASA system called iLab-Edinburgh, using a hybrid approach where the development of their system passes through two stages: the first stage consists of training a set of linear models on lexicon-based word-lemma unigrams. In the second stage, they experimented with different lexica for training the LR models. Their system attained the best performance at a Kendall score of 0.5362. This system was the winner of the Arabic Twitter Task 7 in SemEval 2016.

More recently, [10] explored different deep learning architectures to do Arabic sentiment classification. They used Deep Belief Networks and Deep Auto Encoders combined with features based on an Arabic Sentiment Lexicon. In addition, this system uses other standard lexicon features. To tackle the lack of context handling in the last system, they used another deep learning model called Recursive Auto Encoder (RAE). The experimental results showed that RAE model outperforms all the other models by a large margin of around 9%. RAE model takes advantage from semantic context and parsing order of words. In addition, RAE didn't use any no lexicon, and also no special features, but only raw words as input.

3 Our ASA approach

In this section, we present the main ideas behind our approach for ASA. We begin by introducing the different word representations used in this paper. After that, we do a quick review to the Arabic sentiment datasets used to evaluate our model. Then, we present the main convolutional neural network architecture used on the top of word vectors. Lastly, we show through a series of experiments the importance of using weight normalization for training our neural network to do ASA.

3.1 Word representations

The research in representations of words as continuous vectors has a long history where many ideas were proposed [41,42]. More recently, [43] proposed a model architecture based on feedforward neural networks for estimating neural network language model. The most popular model for word representations was developed by [44] called word2vec where they used either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) model or Skip-Gram (SG) model. Another popular model for word representations developed by [45] called "GloVe" (Global Vectors). The main difference between this model and word2vec models is the representations of a word in vector space: word2vec models use a window approach while GloVe uses the global statistics of word-word co-occurrence in the corpus to be captured by the model.

[46] used word embeddings features for English dependency parsing where they employed flat (non-hierarchical) cluster IDs and binary strings obtained via

sign quantization of the vectors. For chunking, [47] showed that adding word embeddings allows the English chunker to increase its F1-score. [48] showed that adding word embeddings as features for English part-of-speech (POS) tagging task helped the model to increase its performance. [49] argued that using word embeddings in parsing English text improved the system performance.

3.2 ASA Datasets

We have examined the performance of the proposed model on MSA/multi-dialectal textual content. Experiments were conducted on two datasets manually annotated for positive, negative or neutral polarity, they are:

- Arabic Sentiment Tweets Dataset (ASTD): A large sized-dataset of 10006 MSA/dialectal tweets collected and annotated by [50]. In our experiments, we have considered objective tweets as neutral ones. In addition, the set adopted for training was unbalanced.
- SemEval: Represents the Arabic dataset provided for the shared Task 4 entitled "Sentiment Analysis in Twitter " in the international contest of SemEval-2017 [51]. It is a medium-sized dataset of 3355 tweets written in MSA and several Arabic dialects (Egyptian, Syrian, etc.).

Each dataset has been divided into a training set to train the model, a development set to tune it and a test set for evaluation. The detailed statistics of the polarity distribution across these sets are reviewed in Table 1 and Table 2.

Dataset	**Size**	**Positive**	**Negative**	**Neutral**
SemEval	2684	521	1014	1149
ASTD	8005	613	1280	6112

Table 1: The polarity distribution across positive, negative and neutral classes in training sets

Dataset	**Size**	**Positive**	**Negative**	**Neutral**
SemEval	671	222	128	321
ASTD	2001	186	404	1411

Table 2: The polarity distribution across positive, negative and neutral classes in test sets

3.3 Convolutional neural networks Architecture

Convolutional neural networks are powerful machine learning models that shown good performance in many NLP tasks such NER, machine translation, text summarization and language modeling. In this work, we use CNN for ASA. The main architecture of our CNN is depicted in figure 1 where we are classifying the Arabic sentence **شاهد زبون فرنسي غاضب يقتحم متجر شركة ابل تكنولوجيا** which means in English: An angry French customer breaks into the Apple Tech store.

In the first step, we initialize each word in the sentence with pretrained word representations from one of the three models: Glove, Skip-Gram or CBOW. Then, we feed these word vectors to the convolutional neural network as input sentences which will be used in the convolutional layer and max pooling as the next step. Finally, the sentence is classified as "Positive", "Negative" or "Neutral" which can be seen in the last layer (FC with Softmax) containing three output classes. Our CNN architecture is a variant of [33] model where the authors used it to perform part-of-speech tagging, chunking, named entity recognition, and semantic role labeling.

Given a tweet T with length n where we add padding whenever it is necessary for the model, T is represented as the following:

$$v_1^n = v_1 \oplus v_2 \oplus ... \oplus v_n \tag{1}$$

where $v_i \in R^k$ represents the word vector of the $i-th$ word in the sentence S with k is the dimension of the word vector and $\oplus$ represents the concatenation operator. We use successive filters $w \in R^{mk}$ to obtain multiples feature map. Each filter is applied to a window of m words to get a single feature map:

$$F_i = f(w.v_i^{i+m-1} + b) \tag{2}$$

where b is the bias and f is the non-linearity where we used ReLU (Rectified Linear Unit). The general form of a feature map is the following:

$$F = [F_1, F_2, ..., F_{n-m+1}] \tag{3}$$

where $F \in R^{n-m+1}$. In the next step, we applied a max-over-time pooling operation [33] to the feature map and take the maximum value. The results are feed to a fully connected softmax layer to get probabilities over the tweets (Figure 1). We repeat the same operation to many filters to get the next convolutional layer. To fight overfitting and prevent co-adaptation of hidden units, we use Dropout method [52] as the main method for regularization in our model.

4 Experiments

In this section, we begin by presenting the details about the training process and hyperparameters used by our model. After that, we give the experimental results obtained by using our model based on convolutional neural networks, pretrained word representations and weight normalization.

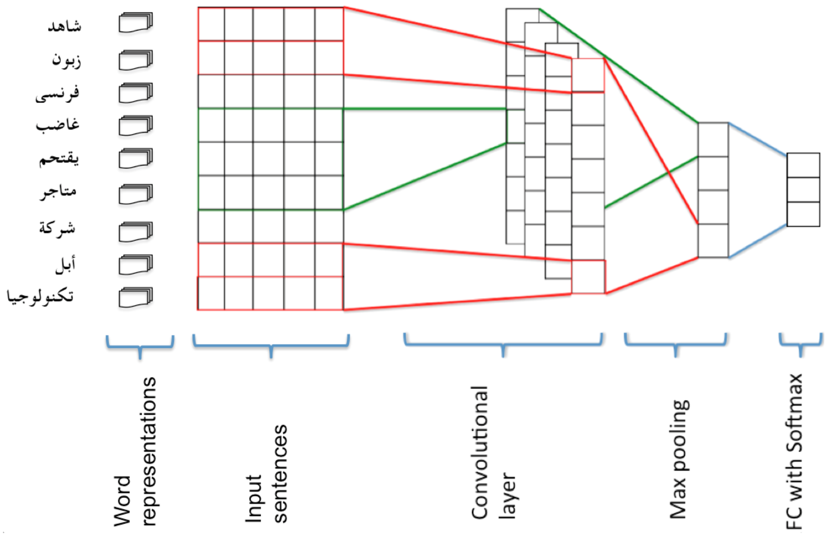

Figure 1: The system architecture. The word vectors are initialized with pre-trained word representations from one of the three models: Glove, Skip-Gram or CBOW. We feed these word vectors to the convolutional neural network as input sentences. Here, the sentences will be classified as "Positive", "Negative" or "Neutral" which can be seen in the last layer (FC with Softmax)

4.1 Training details

To train our convolutional neural network, we use backpropagation algorithm to update our model parameters on every training example using stochastic gradient descent (SGD) over mini-batches. Several methods have been proposed to enhance the performance of SGD, such as Adadelta [53] or Adam [54]. Although we observe faster convergence using these methods, none of them perform as well as SGD.

For other specific CNN hyperparameters, we note that we used the SemEval dev set to choose the best ones. We used ReLU (Rectified Linear Unit) as the non-linearity function, which proved its efficiency. It should be noted that we explored other non)linearity functions such *Sigmoid* function, $Tanh$ function, and others, but none of them were able to be efficient than ReLU. To overcome the problem of model overfitting, we use a dropout rate of 0.5. For filter windows, we set their values to 3, 4 and 5 with 100 feature maps each. We choose the size of the mini-batches to be 50. For the ASTD dataset, it should be noted that we randomly select 10% of the training data to be the dev set.

4.2 Results

The experiments were carried out on ASTD and SemEval 2017 datasets. We begin by presenting the effect of using pretrained word representations compared to the random word vectors on ASA.

Table 3 presents the results of our CNN trained on the top of different pretrained word representations on both ASTD and SemEval datasets. The results are compared to the random word vectors.

Models	F1-score	
	ASTD	SemEval
CNN + Random word vectors	64.15%	58.54%
CNN + Glove vectors	71.83%	62.67%
CNN + Skip-Gram model	72.02%	62.89%
CNN + CBOW model	**72.14**%	**63.00**%

Table 3: Results using three pretrained word vectors: Glove, Skip-Gram and CBOW models and the comparison with randomly initialized word vectors.

From the results showed in Table 3, we conclude that the best F1-score was obtained by using the pretrained word vectors from the CBOW model. Furthermore, we observed that using pretrained word embeddings with all models (Glove, Skip-Gram and CBOW) allowed the model to improve significantly its performance over the model used just randomly initialized word vectors.

Firstly, we improved the F1-score by 7.68 points and 4.13 points respectively on the ASTD and SemEval datasets when we used GloVe pretrained word embeddings. Secondly, we obtained 7.87 points and 4.35 points improvement in F1-score on respectively ASTD and SemEval datasets when use pretrained word embeddings from Skip-Gram model. Lastly, we were able to improve the F1-score by 7.99 points and 4.46 points on respectively ASTD and SemEval datasets by using pretrained word embeddings from the CBOW model.

The last results are consistent with the fact that these pretrained word vectors are universal feature extractors that shown an important results in different NLP applications such named entity recognition and Part-of-speech tagging. To compare our model with the previous state-of-the-art systems, we will use the best results obtained from the combination of CNN and CBOW model.

In the next stage, we compare CNN-ASAWR with the previous best models. We note that for SemEval 2017, until now, we were not able to see the description of the winner system in this shared task competition about Arabic sentiment classification. Table 4 shows the comparison between our model and the winner system called NileTMRG. We were able to outperform their system by 0.95 points in F1-score. Table 5 presents the comparison between our model and [50] system. Their system is based on SVM classifier. Related to the results, we were able to outperform their system by a large margin (9.54 points in F1-score).

As far as we know, we are the first to explore the effect of pretrained word representations on ASA. By combining the power of deep convolutional neural networks with word representations learned from unannotated corpora, we were able to obtain state-of-the-art on two publicly available datasets without resorting to any kind of hand engineering features.

Model	F1-score
Model F1-score NileTMRG	61%
CNN-ASAWR	**63**%

Table 4: Comparison between CNN-ASAWR and NileTMRG which is the winner on SemEval 2017.

Model	F1-score
Nabil et al. (2015)	62.60%
CNN-ASAWR	**72.14**%

Table 5: Comparison between CNN-ASAWR and previous best system on ASTD dataset.

5 Conclusion and Future Work

This paper presents a deep convolutional neural network for ASA that provide the best sentiment classification results on two Arabic sentiment datasets: ASTD and SemEval 2017. We took advantage of using convolutional neural networks in ASA since this model works well in computer vision applications such as image captioning, image classification and others, where it achieved state-of-the-art results. The model truly is end-to-end which means that it does not rely on hand engineering features considered as time consuming.

A key aspect of our model is that it explores the power of deep convolutional neural networks trained on the top of pretrained word representations trained on unannotated corpora. We investigated different models of word representations (CBOW, Skip-Gram and Glove) and compare the results with randomly initialized word vectors. Experimental results showed that the pretrained word representations largely outperformed the random ones with a good margin. This is consistent to the previous results on English sentiment analysis and also to other NLP applications such NER, POS and chunking and confirm the general idea that these pretrained word vectors are universal feature extractors.

In the future work, we will test our model on more Arabic datasets, especially from Arabic dialects such Moroccan, Tunisian, and others, in order to show its stability to various dialects. Furthermore, we will investigate how adding to our model with more features such as part-of-speech tags features, morphological analysis features, and other would influence the results. Finally, we will explore the use of another pretrained word embeddings trained on multilingual Arabic dialects and compare the results with the word embeddings used in this work.

References

1. Nasukawa, T. and Yi, J.: Sentiment analysis: Capturing favorability using natural language processing. In: Proceedings of the 2nd international conference on Knowledge capture, pp. 70–77 (2003).
2. Liu, B.: Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1–167 (2012).
3. Pang, B. and Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1–135 (2008).
4. Calvo, R.A. and D'Mello, S.: Affect detection: An interdisciplinary review of models, methods, and their applications. IEEE Transactions on affective computing, 1(1), 18–37 (2010).
5. Feldman, R.: Techniques and applications for sentiment analysis. Communications of the ACM, 56(4), 82–89 (2013).
6. Ekman, P.: An argument for basic emotions. Cognition & emotion, 6(3-4), pp.169–200 (1992).
7. Mourad, A. and Darwish, K.: Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs. In: Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp. 55–64 (2013).
8. Abdul-Mageed, M., Kuebler, S. and Diab, M.: SAMAR: A system for subjectivity and sentiment analysis of social media Arabic. In: Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), ICC Jeju, Republic of Korea (2012).
9. Refaee, E. and Rieser, V.: iLab-Edinburgh at SemEval-2016 Task 7: A hybrid approach for determining sentiment intensity of Arabic Twitter phrases. In: Proceedings of SemEval-2016, pp.474–480 (2016).
10. Al Sallab, A.A., Baly, R., Badaro, G., Hajj, H., El Hajj, W. and Shaban, K.B.: Deep learning models for sentiment analysis in Arabic. In: ANLP Workshop 2015, p. 9 (2015).
11. Ballesteros, M., Dyer, C. and Smith, N.A.: Improved transition-based parsing by modeling characters instead of words with LSTMs. arXiv preprint arXiv:1508.00657 (2015).
12. Mikolov, T., Kombrink, S., Burget, L., Černocký, J. and Khudanpur, S.: Extensions of recurrent neural network language model. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5528–5531 (2011).
13. Xiong, C., Merity, S. and Socher, R.: Dynamic memory networks for visual and textual question answering. arXiv, 1603 (2016).
14. Chen, D., Fisch, A., Weston, J. and Bordes, A.: Reading Wikipedia to Answer Open-Domain Questions. arXiv preprint arXiv:1704.00051 (2017).
15. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013).

16. Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014).
17. Luong, M.T. and Manning, C.D.: Achieving open vocabulary neural machine translation with hybrid word-character models. arXiv preprint arXiv:1604.00788 (2016).
18. Ma, X. and Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016).
19. Reimers, N., Eckle-Kohler, J., Schnober, C., Kim, J. and Gurevych, I.: Germeval-2014: Nested named entity recognition with neural networks. In: Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition, Hildesheim, Germany, (2014).
20. Bonadiman, D., Severyn, A. and Moschitti, A.: December. Deep Neural Networks for Named Entity Recognition in Italian. In: Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it 2015, p. 51 (2015).
21. Gridach, M.: Character-Aware Neural Networks for Arabic Named Entity Recognition for Social Media. WSSANLP 2016, p.23 (2016).
22. Hochreiter, S. and Schmidhuber, J.: Long short-term memory. Neural computation, 9(8), 1735–1780 (1997).
23. Gers, F.A., Schmidhuber, J. and Cummins, F.: Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451–2471 (2000).
24. Huang, Z., Xu, W. and Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015).
25. Fukushima, K.: Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural networks, 1(2), 119–130 (1988).
26. LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, 86(11), pp.2278–2324 (1998).
27. Ren, S., He, K., Girshick, R. and Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91–99 (2015).
28. Girshick, R., Donahue, J., Darrell, T. and Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587 (2014).
29. Farabet, C., Couprie, C., Najman, L. and LeCun, Y.: Scene parsing with multiscale feature learning, purity trees, and optimal covers. arXiv preprint arXiv:1202.2160 (2012).
30. Vinyals, O., Toshev, A., Bengio, S. and Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156–3164 (2015).
31. Karpathy, A. and Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128-3137 (2015).
32. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J. and Zhang, X.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016).
33. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493–2537 (2011).
34. Kalchbrenner, N., Grefenstette, E. and Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014).

35. Shen, Y., He, X., Gao, J., Deng, L. and Mesnil, G.: Learning semantic representations using convolutional neural networks for web search. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 373–374 (2014).
36. Yih, W.T., Toutanova, K., Platt, J.C. and Meek, C.: Learning discriminative projections for text similarity measures. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 247–256 (2011).
37. Gehring, J., Auli, M., Grangier, D., Yarats, D. and Dauphin, Y.N.: Convolutional Sequence to Sequence Learning. arXiv preprint arXiv:1705.03122 (2017).
38. Paulus, R., Xiong, C. and Socher, R.: A Deep Reinforced Model for Abstractive Summarization. arXiv preprint arXiv:1705.04304 (2017).
39. Zahran, M.A., Magooda, A., Mahgoub, A.Y., Raafat, H., Rashwan, M. and Atyia, A.: Word Representations in Vector Space and their Applications for Arabic. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 430–443 (2015).
40. Abbasi, A., Chen, H. and Salem, A.: Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. ACM Transactions on Information Systems (TOIS), 26(3), p.12 (2008).
41. Hinton, G.E., Rumelhart, D.E. and Williams, R.J.: Learning internal representations by back-propagating errors. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1 (1985).
42. Hinton, G.E., McClelland, J.L. and Rumelhart, D.E.: Distributed representations. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition (1986).
43. Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C.: A neural probabilistic language model. Journal of machine learning research, 3(Feb),1137–1155 (2003).
44. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013).
45. Pennington, J., Socher, R. and Manning, C.D. Glove: Global Vectors for Word Representation. In: EMNLP Vol. 14, pp. 1532–1543 (2014).
46. Hisamoto, S., Duh, K. and Matsumoto, Y.: An empirical investigation of word representations for parsing the web. In: Proceedings of ANLP, pp. 188–193 (2013).
47. Turian, J., Ratinov, L. and Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics, pp. 384–394 (2010).
48. Huang, F., Ahuja, A., Downey, D., Yang, Y., Guo, Y. and Yates, A.: Learning representations for weakly supervised natural language processing tasks. Computational Linguistics, 40(1), 85–120 (2014).
49. Bansal, M., Gimpel, K. and Livescu, K.: Tailoring Continuous Word Representations for Dependency Parsing. In: ACL (2), pp. 809–815 (2014).
50. Nabil, M., Aly, M.A. and Atiya, A.F.: ASTD: Arabic Sentiment Tweets Dataset. In: EMNLP , pp. 2515–2519 (2015).
51. Rosenthal, S., Farra, N., and Nakov, P.: SemEval-2017 task 4: Sentiment analysis in Twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation. Association for Computational Linguistics, Vancouver, Canada, SemEval'17 (2017).
52. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929–1958 (2014).
53. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012).

54. Kingma, D. and Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
55. Russakovsky O. and Deng J. and Su H.and Krause J. and Satheesh S. and Ma S. and Huang Z. and Karpathy A. and Khosla A. and Bernstein M. and Berg A. C.and Fei-Fei L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3), 211–252 (2015).
56. Abdul-Mageed, Muhammad, Diab, Mona T., et Korayem, Mohammed. Subjectivity and sentiment analysis of modern standard Arabic. In : Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2. Association for Computational Linguistics, 2011. p. 587-591.