Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 142 (2018) 339–346

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

Automated Sentence Boundary Detection in Modern Standard Arabic Transcripts using Deep Neural Networks

Carlos-Emiliano González-Gallardo[a,*], Elvys Linhares Pontes[a], Fatiha Sadat[b], Juan-Manuel Torres-Moreno[a,b,c]

[a] *LIA - Université d'Avignon et des Pays de Vaucluse, 339 chemin des Meinajaries, 84140, Avignon, France*
[b] *Université du Québec a Montréal, C.P. 8888, succ. Centre-ville, Montréal (Québec) H3C 3P8 Canada*
[c] *GIGL, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal (Québec) H3C 3A7 Canada*

Abstract

The increased volumes of Arabic sources of data available on the Web has boosted the development of Natural Language Processing (NLP) tools over different tasks and applications. However, to take advantage from a vast amount of these applications, a prior segmentation task call Sentence Boundary Detection (SBD) is needed. In this paper we focus on SBD over Modern Standard Arabic (MSA) by comparing two different approaches based on Deep Neural Networks (DNN) using out-of-domain and in-domain training data with only lexical features (represented as character embedding) while conducting two scenarios based on a Convolutional Neural Network and a Recurrent Neural Network with attention mechanism architectures. While tuning a big out-of-domain dataset with a smaller in-domain dataset, improves the performance in general. Our evaluations were based on IWSLT 2017 TED talks transcripts and showed similarities and differences depending of the SBD method. MSA carries certain complications given its rich and complex morphology. However, using only lexical features for Arabic SBD is an acceptable option when the source audio signal is not available and a certain level of language independence needs to be reached.

Keywords: Sentence Boundary Detection; Speech-to-Text; Transcription; Modern Standard Arabic; Deep Neural Networks

1. Introduction

Arabic language is known to be challenging given its complex linguistic structure [3] and dialect variations [14]. However, the development of Arabic Natural Language Processing (NLP) tools has increased these last years, creating a large set of state-of-the-art applications including POS taggers, syntactic parsers, information retrieval, machine translation, automatic speech recognition and synthesis systems [9, 17]. Some NLP libraries and tools like Python

* Corresponding author. Tel.: +33 04 90 84 35 68.
E-mail address: carlos-emiliano.gonzalez-gallardo@alumni.univ-avignon.fr

1877-0509
10.1016/j.procs.2018.10.485

NLTK[1], OpenNLP[2] [5], UIMA[3] [10], LIMA[4] and NooJ[5] [26], which were originally created for non Arabic texts now include Arabic extensions. By contrast, other libraries have been developed exclusively for Arabic.

Althobaiti *et al.* developed AraNLP [2], which is focused on Arabic preprocessing. This library contains some tools covering tokenization, stemming, POS tagging, sentence detection, word segmentation, normalization and punctuation/diacritic deletion.

MADAMIRA[6], developed by Pasha *et al.*, is an toolkit which combines two previous Arabic NLP systems: MADA[13] and AMIRA[7]. It provides the following NLP tools for Modern Standard Arabic (MSA) and the Egyptian dialect: lemmatization, diacritization, glossing, POS tagging, morphological analysis, morphological disambiguation, stemming, tokenization, base phrase chunking, name entity recognition and word-level disambiguation [23].

The Software Architecture For Arabic language pRocessing (SAFAR) [27] is a framework developed by Souteh *et al.* that aims to gather developed Arabic NLP tools within a single homogeneous architecture and create new ones if necessary. Implemented tools within SAFAR include morphological analyzers, stemmers, syntactic parsers, normalizers, tokenizers, sentence splitters, transliteration tools and question answering applications.

Works have also been made regarding unstructured and noisy Arabic texts, found in social media datasets like microblogs and tweets. Added to the common difficulties of working with structured Arabic texts, Arabic tweets carry other problems like high degree of ambiguity, spelling mistakes, etc. Mallek *et al.* [20] implemented a phrase-based statistical machine translation system from MSA tweets into English. They conclude that a preprocessing step for this kind of noisy texts is very useful to improve the translation results. El-Masri *et al.* [8] presented a sentiment analysis tool over Arabic tweets using a Naive Bayes learning approach. Their model detects the polarity of a group of tweet classifying it in four possible classes: positive, negative, both and neutral.

Access to Internet multimedia platforms nowadays available like Youtube[7], TED[8] and Dailymotion[9] opens a new universe for Arabic NLP tools. Automatic Speech Recognition (ASR) systems can be used to transcribe multimedia content ASR aims to transform spoken data into a written representation, thus enabling natural human-machine interaction with further NLP tasks [31]. Performance of ASR systems for MSA has improve in the last years given the amount of data available for training and testing this systems. Tomashenko *et al.* [28] trained a Deep Neural Network (DNN) with Gaussian Mixture Models derived features and time-delay neural networks for acoustic models over 1128 hours of MSA broadcast speech and 110 million words, reporting a WER of 23%. Menacer *et al.* [21] developed an ASR system for MSA based on the Kaldi toolkit[10] [25], where recognition is achieved using a DNN Hidden Markov model over 63 hours of spoken transcribed data and 1000 million words from the Arabic Gigaword Corpus. They reported a result of 14.42% in terms of WER.

Despite the good performance of modern ASR systems, transcripts do not carry syntactic information as sentence boundaries, which is a major problem for further NLP tasks like POS tagging, automatic text summarization [29], machine translation and sentiment analysis among others. In this paper we address the Sentence Boundary Detection (SBD) task over MSA transcripts to automatically predict or restore the boundaries over transcripts.

The rest of this article is organized as follows. In Section 2 we present an overview of related work concerning SBD. The experimental setup is introduced in Section 3. Experiments and results are addressed in Section 4. In Section 5 Discussion over the methods and difficulties are presented. Finally, Section 6 concludes the paper.

[1] https://www.nltk.org/
[2] https://opennlp.apache.org/
[3] https://uima.apache.org/
[4] https://github.com/aymara/lima/wiki
[5] http://www.nooj-association.org/
[6] https://camel.abudhabi.nyu.edu/madamira/
[7] https://www.youtube.com/
[8] https://www.ted.com/
[9] https://www.dailymotion.com/fr
[10] http://kaldi-asr.org/

2. Sentence Boundary Detection for MSA

Sentence Boundary Detection (SBD) is of vital importance given that in general, ASR systems focus on obtaining the correct sequence of transcribed words with almost no concern of the overall structure of the document, thus lacking of syntactic information [12].

A big complication of segmenting a transcript is the flurry definition of sentence in spoken language. In standard conversations or in simpler scenarios like a monologue, ideas are organized very differently compared to written language. Added to this, Arabic carries other difficulties like word ambiguity, structural ambiguity, lack of punctuation marks, use of connective words and agglutination [15].

To our knowledge no much work has been developed over Arabic SBD. The Arabic Texts Segmentation System or STAr (by its acronym in French), created by Belguith *et al.* [15] is a text segmentation system for Arabic based on a set of rules created from the contextual analysis of punctuation marks and a list of particles which play the role of sentence boundaries. Segmentation consists on the disambiguation of sentence boundaries and paragraphs. Even though they did not report any experiment with transcripts, while it is possible to apply the method over this type of data.

Zribi *et al.* [32] presented three methods for the detection of sentence boundaries in transcribed Tunisian Arabic using lexical and prosodic features. The first method is composed of two sets of handmade rules: 1) based on oral specific lexical items and prosodic features and 2) based on connectors, personal and relative pronouns, verbs, etc. The second method corresponds to a statistical method based on a decision tree algorithm. It classifies a word into four different classes: 1) first word of a sentence, 2) word within a sentence, 3) last word of a sentence and 4) one word sentence. The third method combines the previous two in an hybrid framework.

3. Methodology

3.1. Datasets

One of the objectives of the current research is to analyze the impact of cross-domain datasets during the evaluation phase of SBD. The first dataset (*TED*) corresponds to the Multilingual Task 2017 proposed by The International Workshop on Spoken Language Translation (IWSLT)[11]. It consists of 122 TED talks[12] manually transcribed containing 168,354 words. The second dataset (*GW*) is the Arabic Gigaword[13], which gathers a series of different Arabic news wires containing around 938M words (after XML extraction). We used a sample of 70,261,169 words which corresponds to the Asharq Al-Awsat (aaw_arb) news wire.

As for the preprocessing, we applied a normalization and tokenization process over both datasets. During the normalization phase, all punctuation marks (؟ ? ! ! ، , . . . : ;) were mapped to a common boundary symbol, which corresponds to the "BOUNDARY" class. Based on the experiments performed by Alotaiby *et al.* [1], where they observed that a big lexicon could be reduced about 24.54%, we used the MADAMIRA toolkit to perform a tokenization over both datasets to reduce the dimensionality. The following proclitics and enclitics were separated, generating two or more tokens depending of the amount of clitics within the word:

ال ، و ، ف ، ل ، ب ، ك ، س ، ي ، ا ، ني ، كمَا ، كم ، كن ، ةَ ، همَا ، هن ، هم ، نَا

The final size of the datasets after normalization and tokenization as well as the training, validation and testing distribution can be observed in Table 1. We opted to omit the validation set for *TED* given its reduced size.

[11] http://workshop2017.iwslt.org/
[12] https://www.ted.com/talks?language=en
[13] https://catalog.ldc.upenn.edu/LDC2011T11

It is important to remark the class distribution disparity. Given the nature of the dataset, where sentence boundaries are much less frequent that nonsentence boundaries, the "BOUNDARY" class oscillates between 6.185% and 9.981% (Table 2). Liu *et al.* developed a wide study [19] addressing this disparity behavior with techniques like downsampling, oversampling and replication in a Hidden Markov Model framework. They concluded that depending on the evaluation metric and posterior processing, resampling may or not be useful. Table 1 and 2 show some statistics on the dataset as well as the classes distributions over these datasets.

Table 1. Size and distribution of datasets.

Dataset	*train*	*valid*	*test*	Total
GW	73,608,328	21,030,957	10,515,477	105,154,762
TED	183,314	—	50,881	1942,378

Table 2. Class distribution in percentage (%) over datasets.

	GW			*TED*	
Class	*train*	*valid*	*test*	*train*	*test*
BOUNDARY	6.185	6.519	6.663	9.981	9.699
NO BOUNDARY	93.815	93.481	93.337	90.019	90.301

3.2. Character embeddings

Word representation is an important topic to consider specially for morphology rich languages like Arabic. Common embedding strategies like Word2vec [22] or Glove [24] do not consider the internal structure of words, which is a limitation for morphology rich languages. FastText[14], proposed by Bojanowski *et al.* [6], is a word embedding representation where a vector is associated to each *n*-gram character; therefore, words are represented as the sum of their character vectors.

We opted for FastText vectors to represent our datasets and conduct our experiments given its advantages concerning morphology rich languages. We performed a 300 dimension vector induction with the complete *GW* dataset obtaining 102,248 vectors.

3.3. The Proposed Methods

We are interested in comparing addressed Arabic SBD with two different Neural Network approaches, CNN and RNN.

3.3.1. CNN (SBD_{Conv})

For this method we took into consideration the Convolutional Neural Network (CNN) models proposed by González-Gallardo *et al.* [11] for French SBD. CNN are a type of DNN in which certain hidden layers behave like filters that share their parameters across space. At the last part of the CNN, a set of fully connected layers choose within a set of possible outputs the most probable one. The input layer of a CNN is represented by a $m \times n$ matrix where each cell c_{ij} may correspond to an image's pixel in image processing. For our purpose this matrix represents the relation between a window of m words and their corresponding n dimensional FastText vectors.

The hidden layers inside both CNNs consist of an arrange of convolutional, pooling and fully connected layers blocks. We consider two convolutional layers, with valid padding and stride value of one. The first layer has a 3-shape

[14] https://fasttext.cc/

kernel and 32 output filters while the second has a 2-shape kernel and 64 output filters. To downsample and centralize the attention of the CNN in the middle word of the window, a max pooling layer with 2x3-shape kernel and stride of 1x3 is implemented. The final part of the CNN is formed by 3 fully connected layers with 2048, 4096 and 2048 neurons each and a dropout layer attached to the last layer. RELU activation functions are used to remove linearity of all convolutional, max pooling and fully connected layers.

3.3.2. LSTM (SBD_{LSTM})

Inspired by [18, 30], our second model follows a sequence-to-sequence paradigm using the attention mechanism to verify which words of a sentence c represent a sentence boundary (Figure 1). The words in a sentence c are represented by their FastText embedding. Then, a first LSTM encodes this sentence [16] and a second LSTM with attention mechanism generates the sequence of labels that determine the sentence boundary. The attention mechanism decides which input region to focus in order to generate the next output [4]. LSTM with attention mechanism is composed of input i_t, control state c_t and memory state m_t that are updated at time step t (Equations 2-7).

$$c_t = [we_t, c_t] \tag{1}$$

$$i_t = \text{sigm}(W_1 x_t + W_2 h_{t-1}) \tag{2}$$

$$i'_t = \tanh(W_3 x_t + W_4 h_{t-1}) \tag{3}$$

$$f_t = \text{sigm}(W_5 x_t + W_6 h_{t-1}) \tag{4}$$

$$o_t = \text{sigm}(W_7 x_t + W_8 h_{t-1}) \tag{5}$$

$$m_t = m_{t-1} \odot f_t + i_t \odot i'_t \tag{6}$$

$$h_t = m_t \odot o_t \tag{7}$$

where the operator $\odot$ denotes element-wise multiplication, we_t is the FastText embedding of the word at the time step t, c_t is the context vector, the matrices $W_1, W_2, ..., W_8$ and the vector h_0 are the parameters of the model, and all the non-linearities are computed element-wise. The context vector c_t at time t is calculated as a sum of all hidden states of the encoder weight:

$$c_t = \sum_{j=1}^{T} \alpha_{tj} \cdot h_E^j \tag{8}$$

$$r_{tj} = v_\alpha^T \tanh(W_\alpha h_{t-1} + U_\alpha h_E^j) \tag{9}$$

$$\alpha_{tj} = \text{softmax}(r_{tj}) \tag{10}$$

the probability α_{tj} represents the importance of each hidden state of the encoder h_E^j in the prediction of the current state h_t.

4. Experiments and Evaluations

We conducted two experimental scenarios for SBD over MSA using the neural models described in Section 3.3. Lexical features for both DNN models correspond to the FastText character embeddings described in Section 3.2. Vectors of OOV words from the embedding model are generated from the word's n-grams vectors, eliminating unknown vectors.

The performance is measured as a binary classification task, where the "BOUNDARY" (BOUND) class corresponds to the words followed by a punctuation mark, while the "NO BOUNDARY" (NO_BOUND) class corresponds to those words not followed by a punctuation mark. Given the unbalanced nature of the dataset, a global metric like Accuracy is likely to be biased by the larger class; thus we also compute Precision, Recall and F1 for each class.

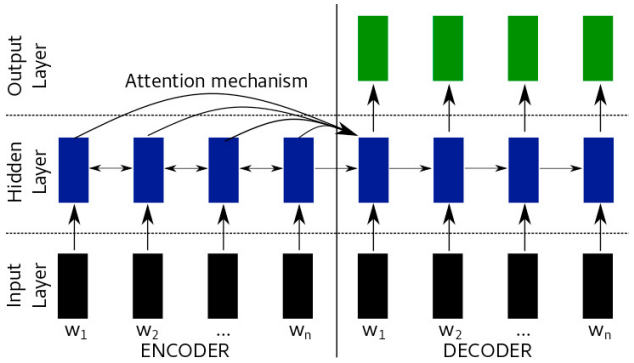

Fig. 1. The words are represented by the word embedding representations. The attention mechanism improves the decode processing. The output layer is composed of by 0 or 1.

4.1. Scenario 1

With this first scenario (S_1) we wanted to observe the impact of applying a SBD model trained with a big dataset collected from written sources over a spoken source dataset. We first conducted training, validation and test on both SBD_{Conv} and SBD_{LSTM} systems with $GW_{train,valid,test}$ for 3 and 7 epochs respectively. Number of epochs were dynamically decided with GW_{valid} before overfitting. Then, we used TED_{test} for evaluating the performance of the trained models over an out-of-domain dataset.

Results for this scenario are shown in Table 3. The high Accuracy (0.963) of SBD_{Conv} over the GigaWord test dataset ($SBD_{Conv(GW)}$) may give an erroneous idea of the model performance. This behavior repeats over the rest of the models. The model is really good predicting the NO_BOUND class a F1 of 0.980. It is important to note almost 40% drop in Recall between both classes, which reflects model's difficulties to retrieve the corresponding instances of the BOUND class. In most values, $SBD_{LSTM(GW)}$ show a similar performance than $SBD_{Conv(GW)}$. The difference concerns the BOUND class Recall (0.327) is 46.57% smaller.

Evaluation of SBD_{Conv} over TED_{test} ($SBD_{Conv(TED)}$) shows a interesting behaviour when compared to $SBD_{Conv(GW)}$. Difference in Precision for both classes, as well as the NO_BOUND class in Recall is very small compared to the Recall BOUND class, which falls about 23.04%. Concerning SBD_{LSTM} for the same test dataset ($SBD_{Conv(TED)}$), the biggest drop in performance also corresponds to the Recall BOUND class, where the difference between them is about 35.47%.

Table 3. S_1 results over *GW* and *TED* evaluation datasets.

Model	Accuracy	Precision		Recall		F1	
		NO_BOUND	BOUND	NO_BOUND	BOUND	NO_BOUND	BOUND
$SBD_{Conv(GW)}$	0.963	0.972	0.797	0.989	0.612	0.980	0.684
$SBD_{LSTM(GW)}$	0.947	0.954	0.729	0.991	0.327	0.972	0.451
$SBD_{Conv(TED)}$	0.934	0.945	0.752	0.983	0.471	0.964	0.579
$SBD_{LSTM(TED)}$	0.914	0.921	0.673	0.989	0.211	0.954	0.321

4.2. Scenario 2

The objective of the second scenario (S_2) is to measure the effect of adding a small in-domain spoken dataset over the models trained on S_1. For this scenario we continued training SBD_{Conv} and SBD_{LSTM} systems with TED_{train}. *TED* dataset size is very small for NN training strategies to consider the creation of a validation set. For this reason, GW_{valid} was used during validation phase and epoch control for both systems. The reduced size of TED_{train} lead to a

fast overfitting behaviour, SBD_{Conv} was trained only for one epoch while X for SBD_{LSTM}. Evaluation was performed over the same dataset of S_1 (TED_{test}).

Table 4 shows the results for S_2. As in S_1, accuracy values are very high given the class unbalanced. Concerning $SBD_{Conv(TED)}$, Precision and Recall for the NO_BOUND class is almost the same with just a small difference of 0.005. Continue training SBD_{Conv} with TED_{train} seems to have a negative impact over the BOUND class Precision (0.687), which is 8.25% lower than in S_1. However, BOUND class Recall improves 39.1%. $SBD_{LSTM(TED)}$ show a similar behavior than $SBD_{Conv(TED)}$, a slight improvement for the BOUND class Recall is present but a decrease for Pecision is present.

Table 4. S_2 results over *TED* evaluation dataset.

Model	Accuracy	Precision		Recall		F1	
		NO_BOUND	BOUND	NO_BOUND	BOUND	NO_BOUND	BOUND
$SBD_{Conv(TED)}$	0.938	0.963	0.687	0.968	0.655	0.966	0.671
$SBD_{LSTM(TED)}$	0.911	0.925	0.597	0.981	0.264	0.952	0.366

5. Discussion

Results for S_1 and S_2 show that unbalanced classes distribution seem to impact SBD_{Conv} and SBD_{LSTM} in a similar degree even both methods follow very different learning techniques. SBD_{Conv} focus its attention on analyzing the words contained in a fixed-sized window, making the boundary prediction independent of the actual position within the transcript. Neverless, this advantage is also a drawback for potentially long sentences given that the method is not able to analyze long contexts.

By contrast, the SBD_{LSTM} is characterized by the analysis of a sequence of words to propose the sentence boundary of this sequence. This approach works best when it analyzes a sequence of words at the beginning of sentences. However, our approach analyzes word sequences that can start in the middle or at the end of sentences, which reduces the performance of predicting sentence boundaries. In addition, long and complex sentences are a challenge to code all the information and to generate a correct sentence boundary for this kind of sentences.

6. Conclusion

In this paper we have studied the impact of using cross-domain datasets during the evaluation phase of two Sentence Boundary Detection systems over Modern Standard Arabic. The obtained results show that tuning a model that was originally trained with a big out-of-domain dataset with small in-domain dataset, in general, improves its performance.

Both methods presented a similar behavior when the spoken language evaluation dataset was used. This may lead to think that spoken MSA maintain the same linguistic structures around SUs than written MSA, but also contains some constructions that written language does not contain.

Our method based on LSTM showed to be less effective compared to the one based on CNNs. We think that letting the method to learn from more epochs will help to at least equal the performance.

As future work we will optimize the training parameters of both methods and increment the number of classes to have the possibility of different boundary types. Also, a hybrid system using bi-LSTM and CNN that will combine the advantages of each method, is among our future research directions.

Acknowledgements

We would like to acknowledge the support of CHISTERA-AMIS ANR-15-CHR2-0001 for funding this research through the Access Multilingual Information opinionS (AMIS), (France-Europe) project.

References

[1] Alotaiby, F., Foda, S., Alkharashi, I., 2010. Clitics in arabic language: a statistical study, in: 24th Pacific Asia Conference on Language, Information and Computation.

[2] Althobaiti, M., Kruschwitz, U., Poesio, M., 2014. Aranlp: A java-based library for the processing of arabic text, in: LREC.

[3] Attia, M., Somers, H., 2008. Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation. volume 279. University of Manchester Manchester.

[4] Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473.

[5] Baldridge, J., 2005. The OpenNLP project. http://opennlp.apache.org/.

[6] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2016. Enriching word vectors with subword information. preprint arXiv:1607.04606 .

[7] Diab, M., 2009. Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking, in: 2nd International Conference on Arabic Language Resources and Tools.

[8] El-Masri, M., Altrabsheh, N., Mansour, H., Ramsay, A., 2017. A web-based tool for arabic sentiment analysis. Procedia Computer Science 117, 38–45.

[9] Farghaly, A., Shaalan, K., 2009. Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing (TALIP) 8, 14.

[10] Ferrucci, D., Lally, A., 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering 10, 327–348. doi:10.1017/S1351324904003523.

[11] González-Gallardo, C.E., Torres-Moreno, J.M., 2018. Sentence Boundary Detection for French with Subword-Level Information Vectors and Convolutional Neural Networks. preprint arXiv:1802.04559 .

[12] Gotoh, Y., Renals, S., 2000. Sentence boundary detection in broadcast speech transcripts, in: ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW).

[13] Habash, N., Rambow, O., Roth, R., 2009. Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization, in: 2nd International Conference on Arabic language resources and tools (MEDAR), Cairo, Egypt, p. 62.

[14] Habash, N.Y., 2010. Introduction to arabic natural language processing. Synthesis Lectures on Human Language Technologies 3, 1–187.

[15] Hadrich, L.B., Baccour, L., Mourad, G., 2005. Star: un système de segmentation de textes arabes basé sur lanalyse contextuelle des signes de ponctuations et de certaines particules, in: TALN'05.

[16] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.

[17] Jaafar, Y., Bouzoubaa, K., 2018. A survey and comparative study of arabic nlp architectures, in: Intelligent Natural Language Processing: Trends and Applications. Springer, pp. 585–610.

[18] Linhares Pontes, E., Huet, S., Torres-Moreno, J.M., Linhares, A.C., 2018. Cross-language text summarization using sentence and multi-sentence compression, in: Silberztein, M., Atigui, F., Kornyshova, E., Métais, E., Meziane, F. (Eds.), Natural Language Processing and Information Systems, Springer International Publishing, Cham. pp. 467–479.

[19] Liu, Y., Chawla, N.V., Harper, M.P., Shriberg, E., Stolcke, A., 2006. A study in machine learning from imbalanced data for sentence boundary detection in speech. Computer Speech & Language 20, 468–494.

[20] Mallek, F., Le, N.T., Sadat, F., 2018. Automatic machine translation for arabic tweets, in: Intelligent Natural Language Processing: Trends and Applications. Springer, pp. 101–119.

[21] Menacer, M.A., Mella, O., Fohr, D., Jouvet, D., Langlois, D., Smaili, K., 2017. An enhanced automatic speech recognition system for arabic, in: Third Arabic Natural Language Processing Workshop, pp. 157–165.

[22] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. preprint arXiv:1301.3781 .

[23] Pasha, A., Al-Badrashiny, M., Diab, M.T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R., 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic., in: LREC, pp. 1094–1101.

[24] Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.

[25] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al., 2011. The Kaldi speech recognition toolkit, in: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, IEEE Signal Processing Society.

[26] Silberztein, M., 2005. Nooj: a linguistic annotation system for corpus processing, in: HLT/EMNLP on Interactive Demonstrations, Association for Computational Linguistics. pp. 10–11.

[27] Souteh, Y., Bouzoubaa, K., 2011. Safar platform and its morphological layer, in: Eleventh Conference on Language Engineering ESOLEC, pp. 14–15.

[28] Tomashenko, N., Vythelingum, K., Rousseau, A., Estève, Y., 2016. Lium asr systems for the 2016 multi-genre broadcast arabic challenge, in: Spoken Language Technology Workshop (SLT), 2016, IEEE. pp. 285–291.

[29] Torres-Moreno, J.M., 2014. Automatic Text Summarization. Wiley and Sons.

[30] Tran, N.T., Luong, V.T., Nguyen, N.L.T., Nghiem, M.Q., 2016. Effective attention-based neural architectures for sentence compression with bidirectional long short-term memory, in: Seventh Symposium on Information and Communication Technology, ACM, New York, NY, USA. pp. 123–130. URL: http://doi.acm.org/10.1145/3011077.3011111, doi:10.1145/3011077.3011111.

[31] Yu, D., Deng, L., 2016. Automatic Speech Recognition. Springer.

[32] Zribi, I., Kammoun, I., Ellouze, M., Belguith, L., Blache, P., 2016. Sentence boundary detection for transcribed tunisian arabic, in: 13th Conference on Natural Language Processing (KONVENS 2016), pp. 323–331.