ISSN 2007-9737

Using Earth Mover's Distance and Word Embeddings for Recognizing Textual Entailment in Arabic

Tarik Boudaa, Mohamed El Marouani, Nourddine Enneya

Ibn-Tofail University, Faculty of Sciences,
Laboratory of Informatics Systems and Optimization,
Morocco

tarikboudaa@yahoo.fr, mohamed.elmarouani@gmail.com, enneya@uit.ac.ma

Abstract. Recognizing Textual Entailment (RTE) is a task of Natural Language Processing (NLP), in which two texts denoted TEXT (T) and HYPOTHESIS (H) are processed by a system to determine whether the meaning of H is inferred (entailed) from T or not. This task is useful for several NLP applications and it has attracted a lot of attention in research. Most of the studies are focused on English as a target language. In this paper, we give an overview of the main studies on Textual Entailment for English and Arabic and we present a new approach to deal with this task for Arabic using a measure of similarity based on Earth Mover's Distance and word embeddings. We experimented with this approach using state of the art Arabic NLP tools and we achieved encouraging results. Although we have applied this approach only to Arabic, its application to other languages is still possible.

Keywords. Recognizing textual entailment (RTE), natural language inference (NLI), Arabic NLP, Earth mover's distance, machine learning

1 Introduction

Textual Entailment (TE) is introduced to promote the development of methods that capture major semantic inference, useful across NLP applications, under a generic and unified framework. In fact, many NLP applications, such as Question Answering, Information Extraction, Summarization, and Machine Translation Evaluation, need a practical model to deal with language variability and inference.

In linguistics, a common definition of entailment [1], states that the Entailment is a relationship between two sentences S1 (the entailing sentence) and S2 (the entailed sentence), in such a way that whenever S1 is true, S2 is also true. The notion of Textual Entailment is close to this definition of entailment in linguistics, however, the Textual Entailment, as a computation and applied task allows for cases in which the inference is highly probable to hold between pairs of texts. The Textual Entailment covers the variability of language expressions and the derivation of new information through reasoning.

This practical and generic oriented view of inference is initially defined as a computational and empirical task by Dagan [2], and thereafter established and completed through the series of benchmarks known as the PASCAL Recognizing Textual Entailment Challenges [2–8]. This definition is stated as follows:

"Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by T - the entailing "TEXT", and H - the entailed "HYPOTHESIS". We say that T entails H if, typically, a human reading T would infer that H is most likely true."

This definition assumes that the entailment is not only determined by what it is announced by the TEXT but also is related to human judgment, which is based on the understanding of the language and background knowledge. However, it is important to notice that the specification of the Recognizing Textual Entailment task requires that the TEXT must be an essential part of the reasoning for inferring the truth of the HYPOTHESIS. Hence, RTE systems are not allowed to assume background knowledge that entails the HYPOTHESIS on its own.

On another side, [9] attempted to formalize the definition of the notion of Textual Entailment by proposing a probabilistic definition:

"*We say that a text T entails a hypothesis H if T increases the likelihood of H being true, that is, if: P (H is true / T) > P (H is true)".*

Operationally, in the RTE task, a computational system (RTE system) accepts as input a pair of texts T (TEXT) and H (HYPOTHESIS) and decides whether H is entailed by T according to the definition given above.

Textual Entailment is applied initially in recognition mode, thereafter The RTE PASCAL challenges have led to more complicated applications of Textual Entailment in search mode [4–6] where the task of the systems becomes extracting, from a given document, all texts that entail a given hypothesis.

In recognition mode, there are two variants of systems, depending on their output, namely:

– The two-way RTE systems: in this case, the system receives as input the (T, H) pair and outputs the label “entails” or “not entails”.

– The three-way RTE systems: these systems handle also the case of contradiction between T and H, their output is one of the labels: “Entails”, “Contradiction” or “Unknown”.

To be consistent with the definitions above, a looser definition of contradiction that more closely matches human intuitions is established [10] :

"*The HYPOTHESIS H of an entailment pair contradicts the TEXT T if a human reader would say that H is highly unlikely to be true given the information described in T*."

The present work concentrates on the Arabic language and proposes a new two-way RTE system for Arabic by employing an approach based on measuring semantic similarity between the TEXT and HYPOTHESIS based on Earth mover's distance and word embedding.

The rest of this paper is organized as follows: Section 2 provides an overview of Textual Entailment works in English and Arabic. Section 3 describes the approach. The evaluation is presented in Section 4. Finally, we conclude in Section 5.

2 Related Work

There have been many studies with various approaches conducted on Textual Entailment. In these following subsections, we review the main approaches used to deal with Textual Entailment in previous works for English and Arabic, and we present Word Mover's Distance (WMD) and highlighting its relationship with Earth mover's distance.

2.1 Main Approaches for English Textual Entailment

Textual Entailment is treated in several works by assuming that it correlates with the similarity that can be captured at different levels depending on the chosen linguistic representation of the text (e.g. Bag of Words, Structured Representation, Logical Representation…). Lexical similarity methods depend on some measures of lexical similarities using lexical resources (e.g. [11, 12]). Some systems approximated the entailment problem as of checking if a good alignment exists between parts of the HYPOTHESIS and parts of the TEXT(e.g. [13, 14]). More structured approaches compute also similarities at the syntactic level by comparing syntactic representations. A common way to perform this comparison is by computing the lowest cost transformation of T’s representation to H’s representation by executing a sequence of elementary edit operations [15–18].

There are significant attempts in the literature to deal with Textual Entailment by transforming the pairs of texts (T, H) into logical representations. In this model, the Textual Entailment is handled in terms of logical entailment. Thus, inference holds between T and H if it holds between their logical representations, plausibly by applying a set of relaxations to make systems robust to errors (e.g. [19–22]). Many systems combine logical based approaches with other techniques such as shallow semantic analysis. Boeing Language understanding engine (BLUE) [23] is an example of this combination.

It uses two strategies implemented by two components on the pipeline. The first one is a logical component and the second is based on a bag of word model that ignore structured representation and acts as a post- process of the

logical component to try to test inference validity for pairs that the logical component can't classify.

Another example of hybrid approaches is that made in [24], where authors extract a set of shallow features based mainly on a set of metrics extracted from a bag of word model, and other features based on a deep semantic analysis technique. Then a machine learning technique based on decision trees is applied to combine features extracted from both methods and produce the final entailment decision.

Machine learning is a crucial element in the architecture of the majority of RTE systems. A dominant approach in the literature casts the problem of RTE as a supervised classification.

Each pair (T, H) to check is represented by a feature vector that incorporates scores of multiple measures applied to the pair at different levels by exploring several aspects of natural language at different levels (e.g. lexical level, syntactic, structural, and semantic level). In order to generate the entailment label, this feature vector is given as input to a classifier trained based on the feature vectors of annotated examples. The definition of the feature space in which we represent the pair (T, H) is the key challenge faced by the machine learning approaches (feature engineering).

With the emergence of the techniques based on Deep Learning applied to NLP, recently promising results were obtained on Recognizing Textual Entailment by exploring Deep Learning techniques and sentence encoding, for instance [25] and [26].

Research in Textual Entailment has led so far to the creation of several datasets for English. For instance: RTE Pascal challenges [2–8]; The Stanford Natural Language Inference (SNLI) [25]; Multi-Gene Natural Language Inference (MultiNLI) [27] and SICK (Sentences Involving Compositional Knowledge) [28]

2.2 Arabic Language and RTE Task

Although the considerable work that has been done in RTE, most studies have tended to focus on English. Particularly, a little work handles this task for Arabic. To the best of our knowledge, the first work have been done in 2011 by [29], it investigates the effectiveness of some existing TE approaches when they are applied to Arabic.

The architecture of the implemented system described with more details in [30] was similar to other existing works for English, however, the implementation of each stage attempted to propose some improvements or adaptations in the aim to deal with the problems raised by Arabic.

The main improvement was in the matching component, by extending Zhang-Shasha's Tree Edit Distance Algorithm [31] to cover also operations on sub-trees instead of operating just on the nodes [32]. Authors proposed also to enhance the results of the preprocessing stage through the combination of outputs of multiple syntactic taggers and parsers [30].

Moreover, [33] aimed to improve the accuracy of an existing Textual Entailment engine by giving attention to negation and polarity. Indeed, negation and polarity components are integrated into a pipelined architecture with the entailment engine. If the output of the former is "entails" then a set of negation rules are checked, if these rules induce a positive decision (entails) then the polarity component is asked to make the final decision.

Furthermore, [34] experimented existing techniques for lexical and semantic matching to the Arabic language by implementing a classical pipeline including a preprocessing component followed by some similarity measures at the lexical and semantic level to produce the final entailment decision.

An approach to deal with Arabic RTE by using a logical representation of the TEXT-HYPOTHESIS pair based on first order logic is that presented in [35]. The problem of Textual Entailment is cast as a binary classification problem, where the feature vector used is formed on some scores calculated using a set of similarity metrics applied to the logical representation of the text.

To address the lack of resource problems faced by the Arabic language, the work presented in [36] tried to use word embeddings for Arabic RTE. The RTE task was presented as a binary supervised classification by experimenting with multiple supervised algorithms using features based mainly on distributional word representations using word embedding. Another approach to deal with Arabic RTE is an attempt to use text alignment [37].

The problem of RTE is transformed into the problem of finding the best alignment between T components and H components.

This alignment is modeled as an assignment problem that consists of finding an optimal weight matching in a weighted bipartite graph. Then, a supervised classifier based mainly on a set of features that measure the quality of the alignment between T and H is asked to assign the final entailment label.

There is little work on the application of Textual Entailment as a subtask in Arabic NLP application. For instance, [38] attempted to exploit Textual Entailment to improve Arabic text summarization. The entailment engine used is based on cosine directional similarity. Moreover, [39] used Textual Entailment to perform the Arabic claim verification task. The Textual Entailment component developed in this work employed a cross-lingual approach and use an Enhanced Sequential Inference Model (ESIM) trained on a large corpus for English and evaluated in XNLI multilingual corpus [40].

Unlike research carried out in this area for English, we did not find significant data for Arabic. To the best of our knowledge, the only resources publically available are the dataset ArbTEDS [41] and XNLI multilingual corpus [40].

2.3 Word Mover's Distance

Word Mover's Distance (WMD) [42] is a distance function that exploits word embeddings, which learn semantically meaningful representations for words, to compute a distance between text documents. It is defined as an optimization problem assuming that the dissimilarity between two text documents can be computed as the minimum amount of distance that the embedded words of one document need to "travel" to reach the embedded words of another document. This distance is cast as a special case of the Earth Mover's Distance [43], a well-known problem that in turn can be cast as a transportation problem for which several solvers have been proposed.

The documents are presented using nBOW representation to represent the weight or the importance of each word. More precisely, let d1 and d2 be two documents after removing their Out-Of-Vocabulary (OOV) words (i.e., their words that not exist in the word embeddings vocabulary). Firstly, a vocabulary V is created of words contained in d1 and d2, and used to transform d1 and d2 on BoW representations (Word frequencies), and then each word frequency in document BoW representation is normalized by the length of the document. A distance matrix M between word vectors of the vocabulary V is constructed using Euclidean distance. Finally, the WMD is computed as the EMD between the nBow representations of d1 and d2 using the distance matrix M, formally:

$$WMD(d1, d2) = EMD(nBoW(d1), nBoW(d2), M). \quad (1)$$

3 Proposed Approach

We propose in this work an approach to recognize Textual Entailment by using a new measure of similarity inspired from Word Mover's Distance (WMD) that we modify and we extend in order to take into account some characteristics of Textual Entailment relation. Otherwise, we combine word embedding and classical knowledge resources while computing the distances between individual words to limit the problem of Out-Of-Vocabulary (OOV) related to gaps in a word embedding. This similarity measure is then used as the main feature to make the decision on the entailment label by a machine learning classifier.

3.1 Preprocessing and Enrichment

In the first step, we apply some basic preprocessing operations to pairs (TEXT, HYPOTHESIS) namely: segmenting the text, normalizing temporal expressions and numbers, annotating named entities, extracting lemmas, and removing stop words.

The result of this stage is formulated as a sequence of components belonging to one of the following types: Named Entity (NE), Temporal Expression, Number, ordinary word. Then the TEXT and HYPOTHESIS are represented as a bag of components with their vector presentations in a word embedding model, and each component c_i is associated with its frequency in the text:

$$(c_1, V(c_1), f_1), (c_2, V(c_2), f_2). \quad (2)$$

In the following, we will refer to the frequency of a component in a text by the term weight.

To limit the effect of the order less property of the standard Bag-of-words model the tokens associated to named entities are grouped as a single component and temporal expressions and numbers are filtered out and handled separately.

Unlike in WMD, in our text representation, we do not normalize the weight of a component by the total weight of all components of the document (text). In fact, generally, T contains more words than H. Thus, if we make this normalization, a same word that appears both in T and H will have a higher importance in H. This can be a good choice to compare the similarity or the equivalence of documents, but this seems not to be the case for an asymmetric relation such as Textual Entailment.

Furthermore, one limitation of word embedding is that in order to have a vector representation of a word, it must be in the vocabulary of the used word embedding model. To limit this problem, we used lemmatization and we enriched words and named entities with a set of equivalents using external knowledge resources. If a component is not found in the word embedding vocabulary, it will be replaced by its equivalent found in the vocabulary.

Indeed, in addition to synonyms extracted from Arabic WordNet [44], we used the free and open knowledge base Wikidata to automatically extract equivalents for named entities. This knowledge base contains data mainly structured as items, each one having a label, a description, a set of aliases, and statements that describe detailed characteristics of an item and consist of a property and a value such as "educated at". We exploited this information, to enrich named entities by their equivalents such as for instance "Donald Trump" can be replaced by "Donald John Trump" or "Donald Trump" and "President of the United States" or even "USA".

3.2 RTE Task as Earth Mover's Distance

In the Textual Entailment task, the TEXT generally contains more information than the HYPOTHESIS. While WMD distance performs total matching between two documents, in our case of Textual Entailment, we considered that the comparison between TEXT and HYPOTHESIS should be conducted by a partial matching, and this partial matching is allowed in Earth Mover's Distance. We correlate the problem of recognizing Textual Entailment to the value of EMD between TEXT and HYPOTHESIS representations. As generally, the total weights of the representations of the TEXT and the HYPOTHESIS are unequal we consider the case where weight flows from the heavier representation (the TEXT) to the lighter one (the HYPOTHESIS) until all weight in the lighter representation has been covered.

A formal definition of the Textual Entailment task as an Earth Mover's Distance problem between the representations of the TEXT and HYPOTHESIS is given in the rest of this section.

Let T and H be, respectively, be the TEXT and the HYPOTHESIS. Each component t_i in T (respectively h_i in H) is given a weight w_i (respectively u_i), in such way, T and H can be defined as follows:

$$T = \{(t_1, w_1), (t_2, w_2), \dots, (t_m, w_m)\}, \quad (3)$$

$$H = \{(h_1, u_1), (h_2, u_2), \dots, (h_n, u_n)\}. \quad (4)$$

Let W and U be the total weight in T and H respectively (i.e. $W = \sum_{i=1}^{m} w_i$ and $U = \sum_{i=1}^{n} u_i$).

And $V(t_i)$ and $V(h_i)$ the vector representations of t_i and h_i respectively in a word embedding model.

According to EMD nomenclature, a flow between T and H is defined as a matrix: $f = (f_{ij}) \in R^{m \times n}$ where f_{ij} represents the flow between t_i and h_i (i.e. the amount of weight at t_i which is matched to weight at h_i).

A flow f is feasible between the TEXT and the HYPOTHESIS iff the following EMD constraints are respected:

$$f_{ij} \geq 0 \quad ; \quad i = 1, \dots . m, j = 1, \dots, n, \quad (5)$$

$$\sum_{j=1}^{n} f_{ij} \leq w_i \quad where \quad 1 \leq i \leq m, \quad (6)$$

$$\sum_{i=1}^{m} f_{ij} \leq u_i \quad where \quad 1 \leq j \leq n, \quad (7)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = min(W, U)\,. \quad (8)$$

The work done by feasible flow f in matching T and H is defined as follows:

$$W(f, T, H) = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \times d\left(t_i, h_j\right). \quad (9)$$

While: $d\left(t_i, h_j\right) = 0$ if t_i and h_j are equivalents, else $d\left(t_i, h_j\right)$ is the Euclidean distance between t_i and h_j vector representations $V(t_i)$ and $V\left(h_j\right)$

Finally, we define our measure for capturing Textual Entailment between T and H as the EMD between T and H but without normalizing by the total weights in H, as given by the formula:

$$EMD(T, H) = min_{f \in F(T,H)}\ W(f, T, H). \quad (10)$$

While *F(T, H)* is the set of all feasible flows between T and H. The measure $EMD(T, H)$ is the main feature used with three other features to make the final entailment label by a machine learning classifier. These features are:

– Number of named entities present in H but not exist in T,
– Number of (normalized) temporal expressions present in H but not exist in T,
– Number of (normalized) numbers found in H but not found in T.

4 Evaluation

4.1 Experiments and Result

To validate our approach we used the Arabic Textual Entailment dataset [41] for evaluation, which comprises 600 (T,H) pairs. This dataset also allowed us to compare the accuracy of our system with other works that used the same data. We preprocessed this dataset using Farasa [45] for segmentation, lemmatization, and named entity recognition, and AraTimex [46] for temporal expressions and numbers normalization. For word embeddings, we employed n-gram model of AraVec [47] useful in our case to retrieve named entities containing more than one word.

Taking into account the small size of this dataset, we proposed two evaluation strategies to avoid biased results and then to have a realistic estimate of the generalization error of the model.

In the first strategy, we split the dataset in such a way that 75% of pairs are used for training and model hyperparameters tuning using Grid Search, while the 25% remaining data is used for testing (held out). In the second strategy, we executed Nested 10 K-Fold cross-validation on the entire dataset to avoid the risk of optimistically biasing our model evaluations and yielding an optimistic score. In this case, first, an inner cross validation is used to tune model parameters using Grid Search and then select the best model. Second, an outer cross validation is employed to evaluate the model selected by the inner cross validation in unseen data.

We conducted our experiments with various classifiers (Extra Trees, Random Forest, AdaBoost, Gradient Boosting, Logistic Regression, SVM) using scikit-learn package. The only results reported are those of the best performing classifier (SVM with RBF kernel). Furthermore, to evaluate the impact of combining knowledge extracted from external resources (Wikidata and WordNet) and the use of word embedding, we considered two configurations of the system in these evaluations:

– Baseline configuration (B.C): In this case, the system uses only word embedding.
– Complete configuration (C.C): In this configuration, we combine the use of external knowledge resources and word embedding.

Additionally, we report results obtained using the same previous evaluation settings, while using WMD instead of our extended measure for Textual Entailment.

Table 1 shows the accuracy results of each evaluation strategy.

To compare with state of the art work, we also reported in the same table, the results obtained by previous systems on the same dataset, these systems are: LR-ALL [36] and ARTESys+ [37] described previously in section 2.

All these results obtained by different evaluation strategies, show the success of our approach based on the Earth Mover's Distance for Textual Entailment. Indeed, the results obtained are comparable or exceed the results obtained

previously by the main previous studies on Arabic Textual Entailment. Additionally, results show clearly that our measure performs better in Textual Entailment than WMD created for measuring document similarity, and enhance the accuracy by about 11% in nested cross validation.

Furthermore, results show the effectiveness of combining word embedding and knowledge extracted from resources such as Wikidata and WordNet to enhance the accuracy of recognizing Textual Entailment. In fact, the configuration C.C outperforms significantly the configuration B.C in all evaluations.

4.2 Error Analysis

In our system, we assume that Textual Entailment is in correlation with the similarity measured using EMD between T and H representations. This measure does not take into account the asymmetric nature of Textual Entailment while computing the distance between components of T and H. For instance, in the following example (a) entails (b) while (b) does not entails (a). Our system will label the pair as Entails either if we consider the first sentence as T or as H because the distance computed between the vector representations of the words "assassinated" and "dead" will be always the same:

(a) Ahmed is assassinated.

(b) Ahmed is dead.

We think that accuracy can be improved if we take into account this characteristic of Textual Entailment. Additionally, the enrichment process implemented using knowledge resources Wikidata and WordNet makes it possible to limit the problem of Out-Of-Vocabulary (OOV) words and not to solve it completely. Thus, we think that it is important to quantify the effect of remaining (OOV) words after enrichment described previously. This effect quantification of OOV should be returned by the system together with $EMD(T, H)$ and it can be used as an important element to judge the confidence in the entailment decision.

Another type of error rises from the text representation used that ignores extra-propositional aspects of meaning such as modality and negation and some important linguistic phenomena not taken into account by the bag-of-word model such as co-reference and syntactic relations between words.

Table 1. Evaluation results

Configuration	Acc. 75%-25% split	Acc. 10-CV	Acc. Nested 10-CV
C.C	**77.33 %**	**76.50 %**	**76.00 %**
B.C	73.33 %	72.83 %	72.66 %
WMD	63.33 %	66.66 %	65.00 %
LR-ALL		76.2%	
ARTESys+		75.84%	

5 Conclusion

We propose in this work an approach to recognize Textual Entailment by using a new measure of similarity adapted to Textual Entailment, inspired from WMD and based on Earth mover's distance. Additionally, we combined word embeddings and classical knowledge resources to reduce the impact of OOV words and then increase the accuracy of Textual Entailment recognition. We used this measure with other features characterizing the difference in numbers, temporal expressions, and named entities between T and H. Then, a supervised Textual Entailment classifier, based on SVM, is used to assign the final entailment label.

Results show the effectiveness of this approach for Arabic, and we think that the accuracy may be improved if we add some global features that give attention to extra-propositional aspects of meaning such as modality and negation.

Although the implementation of this approach is applied to Arabic, it remains a generic approach, which can be applied, to other languages, since word embedding is language independent, and the resources used like Wikidata and WordNet exist for many languages.

References

1. **Chierchia, G. & McConnell-Ginet, S. (2000).** *Meaning and grammar: An introduction to semantics*. MIT press.

2. **Dagan, I., Glickman, O., & Magnini, B. (2006).** The PASCAL Recognising Textual Entailment Challenge. In **Quiñonero-Candela, J., Dagan, I., Magnini, B., & d'Alché-Buc, F. (Éd.)**. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment,* pp. 177–190, Springer Berlin Heidelberg. DOI:10.1007/11736790_9.
3. **Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., & Szpektor, I. (2006).** The second pascal recognising textual entailment challenge. *Proceedings of the second PASCAL Challenges Workshop on Recognising Textual Entailment,* Vol. 6, pp. 6–4.
4. **Bentivogli, L., Clark, P., Dagan, I., & Giampiccolo, D. (2011).** The Seventh PASCAL Recognizing Textual Entailment Challenge. *TAC*. DOI:10.1.1.308.9602.
5. **Bentivogli, L., Clark, P., Dagan, I., & Giampiccolo, D. (2009).** The Fifth PASCAL Recognizing Textual Entailment Challenge. *TAC*. DOI:10.1.1.232.1231.
6. **Bentivogli, L., Magnini, B., Dagan, I., Dang, H.T., & Giampiccolo, D. (2009).** The Sixth PASCAL Recognizing Textual Entailment Challenge. *TAC*.
7. **Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Cabrio, E., & Dolan, B. (2008).** The Fourth PASCAL Recognizing Textual Entailment Challenge. *TAC*.
8. **Giampiccolo, D., Magnini, B., Dagan, I., & Dolan, B. (2007).** The third pascal recognizing textual entailment challenge. *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing,* pp. 1–9. Association for Computational Linguistics.
9. **Glickman, O., Dagan, I., & Koppel, M. (2005).** A lexical alignment model for probabilistic textual entailment. *Machine Learning Challenges Workshop,* pp. 287–298. DOI:10.1.1.100.2750.
10. **De Marneffe, M.C., Rafferty, A.N., & Manning, C.D. (2008).** Finding contradictions in text. *Proceedings of ACL´08: HLT*, pp. 1039–1047.
11. **Castillo, J.J. (2011).** A WordNet-based semantic approach to textual entailment and cross-lingual textual entailment. *International Journal of Machine Learning and Cybernetics*, Vol. *2,* No. 3, pp. 177– 189. DOI:10.1007/s13042-011-0026-z.
12. **Jijkoun, V., de Rijke, M., & et. al. (2005).** Recognizing textual entailment using lexical similarity. *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment,* pp. 73–76.
13. **MacCartney, B., Grenager, T., De Marneffe, M.C., Cer, D., & Manning, C.D. (2006).** Learning to recognize features of valid textual entailments. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics,* pp. 41–48. DOI:10.3115/1220835.1220841
14. **De Marneffe, M.C., Grenager, T., MacCartney, B., Cer, D., Ramage, D., Kiddon, C., & Manning, C.D. (2007).** Aligning semantic graphs for textual inference and machine reading. *Proceedings of the AAAI Spring Symposium,* pp. 468–476.
15. **Kouylekov, M. & Magnini, B. (2005).** Recognizing textual entailment with tree edit distance algorithms. *Proceedings of the First Challenge Workshop Recognising Textual Entailment,* pp. 17–20. DOI: 10.1.1.124.247.
16. **Mehdad, Y. (2009).** Automatic cost estimation for tree edit distance using particle swarm optimization. *Proceedings of the ACL-IJCNLP´09 Conference Short Papers,* pp. 289–292. DOI:10.3115/1667583.1667672.
17. **Mehdad, Y., Cabrio, E., Negri, M., Kouylekov, M., & Magnini, B. (2009).** Using Lexical Resources in a Distance-Based Approach to RTE. *TAC*.
18. **Mehdad, Y., Negri, M., Cabrio, E., Kouylekov, M., & Magnini, B. (2009).** Edits: An open source framework for recognizing textual entailment. *Proc. TAC*.
19. **Akhmatova, E. (2005).** Textual entailment resolution via atomic propositions. *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment,* Vol. 150.
20. **Bos, J. & Markert, K. (2006).** When logical inference helps determining textual entailment (and when it doesn't). *Proceedings of the Second Pascal Rte Challenge,* pp. 26.
21. **Raina, R., Ng, A.Y., & Manning, C.D. (2005).** Robust textual inference via learning and abductive reasoning. *AAAI, Conference: Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, pp. 1099–1105.
22. **Fowler, A., Hauser, B., Hodges, D., Niles, I., Novischi, A., & Stephan, J. (2005).** Applying COGEX to recognize textual entailment. *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment,* pp. 69–72. DOI: 10.1007/11736790_24.
23. **Clark, P. & Harrison, P. (2009).** An Inference-Based Approach to Recognizing Entailment. In *TAC*. DOI:10.1.1.148.7817.

24. **Bos, J. & Markert, K. (2005).** Recognising textual entailment with logical inference. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* pp. 628–635. DOI:10.3115/1220575.1220654.
25. **Bowman, S.R., Angeli, G., Potts, C., & Manning, C. D. (2015).** *A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326*.
26. **Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., & Blunsom, P. (2015).** *Reasoning about entailment with neural attention. arXiv:1509.06664*.
27. **Williams, A., Nangia, N., & Bowman, S.R. (2017).** *A broad-coverage challenge corpus for sentence understanding through inference. arXiv:1704. 05426*.
28. **Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., & Zamparelli, R. (2014).** A SICK cure for the evaluation of compositional distributional semantic models. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14),* pp. 216–223.
29. **Alabbas, M. (2011).** ArbTE: Arabic textual entailment. *Proceedings of the Second Student Research Workshop associated with RANLP´11*, pp. 48–53.
30. **Alabbas, M. (2013).** *Textual Entailment for Modern Standard Arabic* (PhD Thesis). The University of Manchester.
31. **Zhang, K., & Shasha, D. (1989).** Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, Vol. 18, No. 6, pp. 1245–1262. DOI:10.1137/ 0218082
32. **Alabbas, M., & Ramsay, A. (2013).** Natural Language Inference for Arabic Using Extended Tree Edit Distance with Subtrees. *Journal of Artificial Intelligence Research*, Vol. 48, pp. 1–22. DOI:10.1613/jair.3892.
33. **L-Khawaldeh, F.T. (2015).** A Study of the Effect of Resolving Negation and Sentiment Analysis in Recognizing Text Entailment for Arabic. *World of Computer Science & Information Technology Journal*, Vol. 5, No. 7, pp. 124–128.
34. **Khader, M., Awajan, A., & Alkouz, A. (2016).** Textual Entailment for Arabic Language based on Lexical and Semantic Matching. *International Journal of Computing & Information Sciences*, Vol. 12, No. 1, pp. 67–74. DOI:10.21700/ijcis.2016.109.
35. **Ben-Sghaier, M., Bakari, W., & Neji, M. (2018).** *Arabic Logic Textual Entailment with Feature Extraction and Combination.* Intelligent Systems Design and Applications, Springer, pp. 400–409. DOI: 10.1007/978-3-030-16660-1_40.
36. **Almarwani, N. & Diab, M. (2017).** Arabic textual entailment with word embeddings. *Proceedings of the Third Arabic Natural Language Processing Workshop,* pp. 185–190. DOI:10.18653/v1/W17-1322.
37. **Boudaa, T., El Marouani, M., & Enneya, N. (2019).** Alignment Based Approach for Arabic Textual Entailment. *Procedia Computer Science*, Vol. 148, pp. 246–255. DOI:10.1016/j.procs.2019.01.067.
38. **AL-Khawaldeh, F.T., & Samawi, V.W. (2015).** *Lexical Cohesion and Entailment based Segmentation for Arabic Text Summarization (LCEAS)*. Vol. 10.
39. **Ghanem, B., Glavaš, G., Giachanou, A., Paolo, S., Ponzetto, P.R., & Rangel, F. (2019).** UPV-UMA at CheckThat! Lab: Verifying Arabic Claims using a Cross Lingual Approach. *CLEF.*
40. **Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S.R., Schwenk, H., & Stoyanov, V. (2018).** *Evaluating Cross-Lingual Sentence Representations.* arXiv:1809.05053.
41. **Alabbas, M. (2013).** A Dataset for Arabic Textual Entailment. *Proceedings of the Student Research Workshop associated with RANLP´13,* pp. 7–13.
42. **Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015).** From word embeddings to document distances. *CML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning,* Vol. 37, pp 957–966. DOI:10.5555/3045118.3045221.
43. **Rubner, Y., Tomasi, C., & Guibas, L.J. (1998).** A metric for distributions with applications to image databases. *Sixth International Conference on Computer Vision,* pp. 59–66. DOI:10.1109/ICCV. 1998.710701.
44. **ElKateb, S., Black, W., Rodríguez, H., Alkhalifa, M., Vossen, P., Pease, A., & Fellbaum, C. (2006).** Building a WordNet for Arabic. *LREC,* pp. 29–34.
45. **Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016).** Farasa: A Fast and Furious Segmenter for Arabic. *Association for Computational Linguistics,* pp. 11–16.
46. **Boudaa, T., El Marouani, M., & Enneya, N. (2018).** Arabic Temporal Expression Tagging and Normalization. *International Conference on Big Data, Cloud and Applications,* pp. 546–557.
47. **Soliman, A.B., Eissa, K., & El-Beltagy, S.R. (2017).** Aravec: A set of Arabic word embedding models for use in Arabic NLP. *Procedia Computer Science*, Vol. 117, pp. 256–265. DOI:10.1016/ j.procs.2017.10.117.

ISSN 2007-9737

Article received on 13/05/2020; accepted on 05/10/2020.
Corresponding author is Tarik Boudaa.