Khloud Al Jallad
Nada Ghneim

# ARNLI: ARABIC NATURAL LANGUAGE INFERENCE ENTAILMENT AND CONTRADICTION DETECTION

**Abstract** *Natural language inference (NLI) is a hot research topic in natural language processing; contradiction-detection between sentences is a special case of NLI. This is considered to be a difficult NLP task that has a significant influence when added as a component in many NLP applications (such as question-answering systems and text summarization). The Arabic language is one of the most challenging low-resource languages for detecting contradictions due to its rich lexical semantic ambiguity. We have created a data set of more than 12k sentences and named it ArNLI; it will be publicly available. Moreover, we have applied a new model that was inspired by Stanford's proposed contradiction-detection solutions for the English language. We proposed an approach for detecting contradictions between pairs of sentences in the Arabic language using a contradiction vector combined with a language model vector as an input to a machine-learning model. We analyzed the results of different traditional machine-learning classifiers and compared their results on our created data set (ArNLI) and on the automatic translation of both the PHEME and SICK English data sets. The best results were achieved by using the random forest classifier, with accuracies of 0.99, 0.60 and 0.75 on PHEME, SICK, and ArNLI respectively.*

**Keywords** textual entailment, Arabic NLP, contradiction detection, contradiction Arabic data set, textual inference

**Citation** Computer Science 24(2) 2023: 187–209

## 1. Introduction

Natural language inference (NLI) is the task of determining whether a given hypothesis can be inferred from a given premise. This task, formerly known as recognizing textual entailment (RTE), has long been a popular task among researchers [41]. As an improvement over the simple binary entailment vs non-entailment scenario, three-way RTE has appeared and is commonly used (entailment, contradiction, neutral [Unknown]). The entailment relationship between two text fragments holds whenever a claim that is present in Fragment B can be concluded from Fragment A. The contradiction relationship applies when a claim in A and a claim in B cannot be true together. The neutral relationship applies when neither A and B entail nor contradict each other. The main impact is that RTE can transfer a problem from text data set language processing to algebra sets and logical implications; for this reason, RTE has a significant influence when added as a component in many NLP applications, as it can simplify problems. Textual inference is a key capability for improving the performance of a wide range of NLP tasks [43], such as question-answering systems [42], Information Retrieval (IR) and Information Extraction (IE) [1], text summarization,[2] next-generation information retrieval [42], machine reading [12, 36], machine translation [37], Natural Language Understanding (NLU) [51], anaphora resolution [13], and argumentation mining [29]. Since 2005, several challenges have been coordinated with the aim to provide concrete data sets that the research community could use to test and compare their different approaches to recognizing entailments. However, RTE from Arabic text remains under-explored. The Arabic Language is one of the most challenging low-resource languages in detecting contradictions due to its lexical richness and semantic ambiguity. Moreover, there is no available benchmark for the contradiction-detection task in the Arabic language to the best of our knowledge. In this paper, we introduce a new high-quality data set for the NLI task for the Arabic language. This data set (named ArNLI) includes more than 6000 pairs of sentences that are annotated in three-way relationship classes (entailment, contradiction, and neutral), where:

- **Contradiction** indicates contradictions between pairs of texts that involve all types that De Marneffe et al. discussed in [34] (antonym, negation, numeric, factive, structure, lexical, WK).
- **Entailment** indicates that two texts entail the same meaning.
- **Neutral** indicates that there is no relationship between two texts.

Using different language-modeling approaches (including word embedding) and the features of different language levels (lexical, semantic), we evaluate different traditional classification models (support vector machine [SVM], stochastic gradient de-

[1] NIST PASCAL Recognizing Textual Entailment Challenge (RTE-5) at TAC 2009: https://tac.nist.gov/2009/RTE/

[2] NIST, 6th Textual Entailment Challenge @ TAC 2010 Knowledge Base Population Validation Pilot Task Guidelines,
TAC Workshop, 2010

scent [SGD], decision tree [DT], ADABoost, k-nearest neighbors [KNN], and random forest [RF]) and compare the results with the translation of famous English benchmarks (due to the lack of benchmarks in Arabic). The rest of the paper is organized as follows: Section 2 will cover the related literature. Section 3 will present our methodology in detail and will describe our created Arabic RTE data set. Section 4 will then discuss the experimental results. Finally, we conclude with future research directions in Section 5.

## 2. Related works

In the recent past, natural language inference (NLI) (formerly known as RTE) has gained significant attention – particularly given its promise for downstream NLP tasks [36]. The majority of research that has been done in NLI has focused on two-way RTE (the simple binary entailment vs non-entailment scenario), whereas three-way RTE (entailment, contradiction, neutral [Unknown]) that focuses on contradiction has been featured in very few research projects. Recent statistics[3] show that research in RTE focuses on big data sets using deep-learning models with transformers such as BERTNLI, RoBERTa, XLNET, and DeBERTa. Thus, most progress in NLI has been limited to English due to the lack of reliable data sets for most of the world's languages. In other languages, different research works have attempted to create data sets for NLI (such as Japanese [19], Chinese [21], Portuguese [40], Italian [9], German [16], Brazilian [17], Persian [6], and Turkish [11]).

As for the Arabic language, an Arabic data set for RTE[4] exists; however, it converts two-way RTE and has only 600 pairs, which is not considered to be enough for any deep-learning methodology. In the rest of the related work section, we will emphasize that three-way RTE that focuses on contradiction is featured in relatively few works, so this is our research interest in this paper.

### 2.1. Related works on English language

Harabagiu et al. [18] presented the first empirical results for contradiction detection (CD) as a task of entailment recognition; however, they focused on specific kinds of contradictions and described a framework for detecting contradictions between sentences. The work had three basic types of linguistic information: (a) negation; (b) relational and modality features, and (c) semantic information. The authors created two corpora for evaluating their system; one was constructed via negating each entailment in the RTE2 data,[5] generating a balanced data set (LCC1 negation data set). To avoid overtraining, negative markers were also added to each non-entailment, making sure that they were not contradictions. The other corpus was created by paraphrasing the hypothesis sentences from LCC-negation to remove the

[3] https://paperswithcode.com/sota/natural-language-inference-on-rte
[4] Arabic textual entailment data set http://www.cs.man.ac.uk/~ramsay/ArabicTE/
[5] Negation data sets, Stanford https://nlp.stanford.edu/projects/contradiction/

negations (LCC-paraphrase). They achieved accuracies of 75.63% on LCC-negation and 62.55% on LCC-paraphrase.

In [34], Rafferty and Manning proposed an appropriate definition of a contradiction for NLP tasks and developed a corpus from which they constructed a typology of contradictions. They found two primary categories of contradiction: (1) those occurring via antonym, negation, and date/number mismatch (which are relatively simple to detect), and (2) contradictions that arise from the use of factive or modal words, structural and subtle lexical contrasts, and world knowledge (WK). They considered contradictions in the first category 'easy' and can be obtained by using existing resources and techniques (e.g., WordNet,[6] VerbOcean). However, contradictions in the second category were considered to be more difficult to detect automatically because they required precise models of sentence meaning. Moreover, they proposed a system that was based on the architecture of the Stanford RTE system [32]; however, they introduced a stage for event co-reference decisions. The features that were used were polarity features, number, date, and time expression features, antonym features, structural features, factivity corpora; one was based on an RTE data set, and the other on 'real life' data. As RTE data sets are balanced between entailments and non-entailments, the RTE3-test data was annotated by NIST as part of the RTE3 pilot task[7] in which systems classify pairs as entailed, contradictory, or neither. As for the real-life corpus,[8] they collected 131 contradictory pairs: 19 from Newswire (mainly looking at related articles in Google News), 51 from Wikipedia, 10 from the Lexis Nexis database, and 51 from the data that was prepared by LDC for the distillation task of the DARPA GALE program. Despite the randomness of the collection, they argued that this corpus may best reflect naturally occurring contradictions.

Ritter et al. [39] proposed contradiction detection using functions (e.g., BornIn [Person] = Place) and a domain-independent algorithm that automatically detected sentences that denoted functions. Their work was based on de Marneffe et al.'s work with a number of modifications. They suggested that global world knowledge is important for constructing a domain-independent system. Moreover, they automatically created a large corpus of obvious contradictions that can be found in arbitrary web text. As for system evaluation, they used the 1000-most-frequent relationships that were extracted by the TextRunner system [52] – 75% were indeed functional. They labeled each of these 8844 pairs (by hand) as contradictory or not.

In [27], Li et al. used a CNN-based (convolutional neural network) model to learn global and local semantic relationships from sentences. They used contradiction-specific word embedding (CWE). CWE was learned from a training corpus that was automatically generated from a paraphrase database and was used as a feature to

[6] WordNet https://wordnet.princeton.edu/

[7] RTE3-pilot, Stanford, 2007 https://nlp.stanford.edu/RTE3-pilot/

[8] Negation Real Life Corpus, Stanford, https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

implement contradiction detection in the SemEval 2014 benchmark data set.[9] The shallow features that were extracted were the number of negation words, the difference of word order, and unaligned words. The experimental results showed optimization on traditional context-based word embedding in contradiction detection, as it improved in accuracy from 75.97 to 82.08% in the contradiction class.

In [45], Sulea proposed applying three-way RTE in social media. The author worked on 5000 pairs that were collected from Twitter to distinguish between tweets that entailed or contradicted each other or that claimed unrelated things. They used neural networks and compared their results on word embeddings with the results that were previously obtained using classical "feature engineering" methods.

Lingam et al. [28] proposed an approach for detecting three different types of contradictions (negations, antonyms, and numeric mismatches) using neural networks and deep learning. They used long short-term memory (LSTM) and global vectors for word representation (GloVe).[10] There were three feature combinations: manual features (Jaccard coefficient, IsNegation flag, IsAntonym flag, and Overlap coefficient), LSTM-based features, and a combination of manual and LSTM features. They did experiments on three publicly available data sets: Stanford, SemEval,[11] and PHEME[12] [26]. In addition, they constructed a data set and made it publicly available. They achieved a 96.85% accuracy for the contradiction class on the PHEME data set.

Moreover, many research papers have applied NLI over the last few years for special domains or optimizing solutions for other complex NLP tasks. For example, Microsoft created a corpus named Microsoft Research Paraphrase Corpus (MRPC) that consists of 5801 sentence pairs that were collected from Newswire articles. Each pair was labeled by human annotators as a paraphrase or not. [14, 35]

Wang et al. [49] proposed GLUE[13] (general language understanding evaluation benchmark) – a tool for evaluating and analyzing the performance of models across a diverse range of existing NLU tasks based on NLI. Moreover, Wang et al. [48] proposed SuperGlue [46] as an improvement on Glue by having more challenging tasks, more diverse task formats, and so on. Glue and SuperGlue contained the QNLI (question-answering NLI) data set, which was a natural language inference data set that was automatically derived from Version 1.1 of the Stanford question-answering data set (SQuAD). SQuAD v1.1 consisted of question-paragraph pairs where one of the sentences in a paragraph (drawn from Wikipedia) contained the answer to the corresponding question (written by an annotator).

The Allen Institute for Artificial Intelligence's research created abductive natural language inference (alphaNLI) [7] that was a common-sense benchmark data set that

[9] SemEval2014 http://alt.qcri.org/semeval2014/

[10] GloVe, Stanford https://nlp.stanford.edu/projects/glove/

[11] SemEval2014 http://alt.qcri.org/semeval2014/

[12] Pheme, 2016 https://www.pheme.eu/2016/04/12/pheme-rte-dataset/

[13] Glue, 2018 https://gluebenchmark.com/

was designed to test an AI system's capability to apply abductive reasoning and common sense to form possible explanations for a given set of observations. Formulated as a binary-classification task, the goal was to pick the most plausible explanatory hypothesis given two observations from a narrative context.

Yuta et al. [25] proposed a data set for NLI at the document-level to automatically support the contract-review process. They simplified the problem by modeling it as multi-label classification over spans instead of trying to predict the start and end tokens, and they showed that Span NLI BERT outperformed the existing models.

Wang et al. [50] solved many NLU tasks by transforming them into NLI tasks; systematic evaluations on 18 standard NLP tasks showed that it improved the various existing SOTA few-shot learning methods by 12% and yielded a competitive few-shot performance with 500-times-larger models (such as GPT-3).

Liu et al. [30] proposed RoBERTa (which was a BERT-tuned model) that achieved state-of-the-art results on GLUE, RACE, and SQuAD without multi-task fine-tuning for GLUE nor additional data for SQuAD.

He et al. [20] proposed the DeBERTa model architecture (decoding-enhanced BERT with disentangled attention) that improved the BERT and RoBERTa models by using two novel techniques (disentangled attention, and an enhanced mask decoder). Compared to RoBERTa-Large, a DeBERTa model that was trained on half of a training data performed consistently better on a wide range of NLP tasks, achieving improvements on MNLI by +0.9% (90.2 vs. 91.1%), on SQuAD v2.0 by +2.3% (88.4 vs. 90.7%), and on RACE by +3.6% (83.2 vs. 86.8%).

## 2.2. Related works on Arabic language

In the Arabic language, only a small amount of research has been done in the RTE domain. Textual entailment in Arabic faces various challenges due to the features of the language [5, 8, 23]. One of these challenges is lexical ambiguity, which is the difficulty of processing texts with missing diacritics. Another challenge is the language's richness in synonyms, where more than a one-word surface may have the same meaning. In addition, Arabic still lacks the large-scale handcrafted computational resources that are very practically used in English (such as a large WordNet). On the other hand, the lack of a large entailment data set has resulted in a lack of deep-learning research experiments (only traditional machine-learning methods have been proposed). Alabbas [3] developed the ArbTE system to evaluate the existing text-entailment techniques when applied to the Arabic language. In the next step, Alabbas suggested extending the basic version of the tree edit distance (TED) algorithm in [4] in order to enhance the matching algorithm to identify TE in Arabic. The author also created a publicly available data set for Arabic textual entailment – ArbTEDS[14] – which consisted of 618 text-hypothesis pairs that were collected from Arabic news websites or annotated pairs that were collected by hand.

[14] Arabic textual entailment data set http://www.cs.man.ac.uk/~ramsay/ArabicTE/

AlKhawaldeh et al. [2] concluded that Arabic entailment accuracy can be enhanced by resolving the negation for the entailment relationship, analyzing the polarity of a text-hypothesis pair, and determining the polarity of the text-hypothesis pair (positive, negative, or neutral). They achieved an accuracy of 69% on the ArbTEDS data set.

Almarwani et al. [5] applied SVM and random forest classifiers to detect RTE in Arabic using word embeddings to overcome the lack of explicit lexical overlaps between T and H sentence pairs. They derived word-vector representations for about 556K words. Other features that were used included similarity scores, named entities, the number of unique instances in T, the number of unique instances in H, the number of unique instances that were in T but not in H (and vice versa), and the number of instances that were in both H and T. All of the features were calculated at the token, lemma, and stem levels. The system achieved an accuracy of 76.2% on the ArbTEDS data set.

Boudaa et al. [10] used a support vector machine algorithm to detect the RTE for the Arabic language. The following analyses were used in the pre-processing stage: named entities, temporal expressions, number/countable pairs, and ordinary words (or sequences of ordinary words). They extracted alignment-based features to find optimal weight matching in a weighted bipartite graph. The system achieved an accuracy of 75.84% on the ArbTEDS data set.

Khader et al. [23] applied a lexical analysis technique of textual entailment for the Arabic language. They added a semantic matching approach to enhance the precision of their system. Their lexical analysis was based on calculating word overlaps and bigram extraction and matching. They combined semantic matching with word overlaps to increase the accuracy of the word matching. They achieved 68 and 58% precision for entails and not-entails, respectively, with an overall recall of 61% on the ArbTEDS data set.

## 3. Our methodology

In this work, we created a data set and proposed a system for detecting NLI in Arabic sentences where the target labels were entailment, contradiction, and neutral (no semantic relationship). Our system consists of three main parts: text pre-processing (cleaning, tokenization, stemming), feature extraction (contradiction feature vector and language model vectors), and the machine-learning model. Figure 1 shows our experimental schema. We will discuss each step in detail in the following subsections.

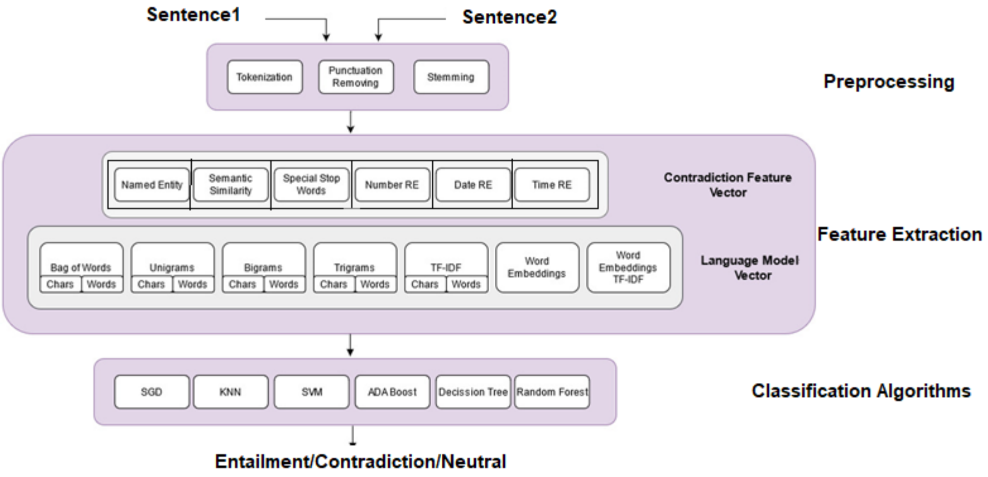

**Figure 1.** Our experimental schema

## 3.1. Our data set

To the best of our knowledge, there is no available Arabic three-way natural language inference (NLI) data set. In order to build our data set, we started by translating two English RTE data sets: the SICK data set [33] (which was used in SemEval_2014_-Task1),[15] and the PHEME data set. The SICK data set consisted of 10,000 English sentence pairs, each annotated for relatedness in meaning and entailment relationship, while the PHEME data set contained 5400 RTE annotated pairs from social media. We named these automatically translated Arabic data sets Ar_SICK and Ar_PHEME, respectively. After automatically translating the two data sets, we selected a subset of the annotated pairs and manually corrected their translations. We augmented this subset with manually translated/annotated pairs from pre-existing sources. Our final Arabic natural language inference (NLI) data set[16] (ArNLI) contained 6366 pairs that were divided as 1932 entailment, 1073 contradiction, and 3361 neutral. The data set was collected as follows:

- **5948 pairs** of AR_SICK data set sentences that were semi-automatically translated and corrected (1714 entailment, 895 contradiction, and 3339 neutral pairs);
- **312 pairs** of ArbTEDS corpus[17] from which we had to re-annotate its sentence classes (entails, not-entails) into the three-way RTE classes that were considered in this study (194 entailment, 113 contradiction, and 5 neutral pairs);
- **35 pairs** of Stanford real-life contradiction corpus [34], which was manually translated (0 entailment, 35 contradictions, and 0 neutral pairs);

[15] semeval2014 task1 2014 https://alt.qcri.org/semeval2014/task1
[16] ArNLI https://github.com/Khloud-AL/ArNLI
[17] Arabic textual entailment data set http://www.cs.man.ac.uk/ ramsay/ArabicTE/

- **71 pairs** of manually annotated sentences (collected from online websites teaching Arabic contradiction, poems, idioms, and paraphrased pairs of Ar_PHEME data set) with 24 entailments, 30 contradictions, and 17 neutral pairs.

  The key statistics of our created data set (ArNLI) are shown in Table 1.

**Table 1**
Key statistics of ArNLI data set

| **Data Size** | |
|---|---|
| Training pairs | 5092 |
| Testing pairs | 1274 |
| **Avg. Sentence Length in tokens** | |
| Hypothesis | 6.623 |
| Premise | 7.246 |
| **Max. Sentence Length in tokens** | |
| Hypothesis | 26 |
| Premise | 57 |

## 3.2. Text preprocessing

In this step, we first tokenized the sentences and removed all of the punctuation marks. To extract the morphological units, we used Snowball Stemmer (which is also known as the Porter2 stemming algorithm) Table 2 presents examples of each step in pre-processing stage.

**Table 2**
Examples of output of each step in preprocessing stage

| **Stage** | **Sentence 1** | **Sentence 2** |
|---|---|---|
| | عملنا في هذا البحث على فهم علاقات الاستدلال و استخراجها بين الجمل في جميع اللغات، وليس فقط اللغة العربية .. | عملنا في هذا البحث على اكتشاف علاقات الاستدلال و التناقضات بين الجمل في اللغة العربية فقط، لم نعمل على اكتشافها في باقي اللغات ! |
| **Tokenization** | (['..', 'عملنا', 'في', 'هذا', 'البحث', 'على', 'فهم', 'علاقات', 'الاستدلال', 'و', 'استخراجها', 'بين', 'الجمل', 'في', 'جميع', 'اللغات،', 'وليس', 'فقط', 'اللغة', 'العربية']) | (['عملنا', 'في', 'هذا', 'البحث', 'على', 'اكتشاف', 'علاقات', 'الاستدلال', 'و', 'التناقضات', 'بين', 'الجمل', 'في', 'اللغة', 'العربية', 'فقط،', 'لم', 'نعمل', 'على', 'اكتشافها', 'في', 'باقي', 'اللغات', '!']) |
| **Punctuation Removal** | ['عملنا', 'في', 'هذا', 'البحث', 'على', 'فهم', 'علاقات', 'الاستدلال', 'و', 'استخراجها', 'بين', 'الجمل', 'في', 'جميع', 'اللغات،', 'وليس', 'فقط', 'اللغة', 'العربية'] | ['عملنا', 'في', 'هذا', 'البحث', 'على', 'اكتشاف', 'علاقات', 'الاستدلال', 'و', 'التناقضات', 'بين', 'الجمل', 'في', 'اللغة', 'العربية', 'فقط،', 'لم', 'نعمل', 'على', 'اكتشافها', 'في', 'باقي', 'اللغات'] |
| **Snowball Stemmer** | ['عمل', 'في', 'هذا', 'بحث', 'على', 'فهم', 'علاق', 'استدلال', 'و', 'استخراج', 'بين', 'جمل', 'في', 'جميع', 'اللغ', 'ليس', 'فقط', 'اللغ', 'عرب'] | ['عمل', 'في', 'هذا', 'بحث', 'على', 'اكتشاف', 'علاق', 'استدلال', 'و', 'تناقض', 'بين', 'جمل', 'في', 'اللغ', 'عرب', 'فقط', 'لم', 'نعمل', 'على', 'اكتشاف', 'في', 'باق', 'اللغ'] |

## 3.3. Feature extraction

In our proposed model, we used different types of features: named entity features, WordNet::Similarity features, special stopword feature, and number, date, and time features. We used different language models such as TFIDF, n-grams, and word embeddings.

### 3.3.1. Contradiction vector's proposed features

### A. Arabic named entity features

Two sentences with different named entities can cause a contradiction in meaning even they may contain most of the same words. For example, the capital of a country is a specific city that cannot be replaced by another city:
ليون عاصمةفرنسا **Contradicts** باريس عاصمةفرنسا
(Paris is the capital of France) **Contradicts** (Lyon in the capital of France)
We used AQMAR[18] to detect the named entities in the sentences. We encoded the values in order to consider three different cases:

- if same ANEs are used in both sentences;
- if different ANEs are used in both sentences;
- if neither sentence contains ANEs.

### B. Semantic similarity features

To be able to focus on the concepts (not merely on the exact words), we added some semantic features that were based on the WordNet::Similarity project. A word can have different meanings according to its context, and this has a direct effect on the relationships between phrases. Semantic-similarity features are calculated for all of the words in Sentence 1 with all of the words in Sentence 2. The similarity features are synonym word count, neutral word count, and antonym word count. Table 3 presents examples of different relationships between sentences in Arabic.

**Table 3**
Different relationship examples between sentences

| Relationship | Sentence 1 | Sentence 2 |
|---|---|---|
| **No relationship with** | شرب الغزال من العين<br>The deer drank from the river | أعاني من حساسيَّة العين<br>I have an eye allergy |
| **Entailment** | اشترى أحمد بيت أمجد<br>Ahmad bought Amjad's house | باع أمجد بيته لأحمد<br>Amjad sold his house to Ahmed |
| **Entailment** | أفل القمر<br>The Moon sets | أشرقت الشَّمس<br>The Sun rises |
| **Contradiction** | أفل القمر<br>The Moon sets | غربت الشَّمس<br>The Sun sets |
| **Contradiction** | لم ينطق بكلمة<br>He did not say a word | قال ماما لقد عدت للمنزل<br>He said, "Mom I am home" |

### C. Arabic special stopword features

Some Arabic stopwords affect the meanings of sentences and, thus, must be considered when studying entailments. In contradiction, for example, negations such as (ما-لا-ليس) and exceptions such as (إلا-سوى-عدا) can alter the results. Moreover, some negation words would mean confirmation if they come together with a negation word, such as (لا-إلا). Table 4 presents some examples of contradictions and entailments using

[18] https://www.cs.cmu.edu/~ark/ArabicNER/

stopwords. In our system, each special stopword will be extracted; then, we will encode the feature values to consider three different cases:

- if special stopword exists in one sentence;
- if special stopword exists in both sentences;
- if special stopword does not exist in either sentence.

**Table 4**
Relationship examples using stopwords

| Relationship | Sentence 1 | Sentence 2 |
|---|---|---|
| **Contradiction** | لا إله<br>No God | لا إله إلا الله<br>No God except Allah |
| **Entailment** | لا يعلم المستقبل إلا الله<br>No one know the future except Allah | الله يعلم المستقبل<br>Allah knows the future |

**D. Number, date, and time features**
We extract those features that concern number, date, and time by using regular expressions to detect patterns. We also take Arabic words that compare quantities into consideration (.cte ,naht erom/يزيد عن ,naht ssel/ينقص ,ylraen/حوالي :xe). Table 5 shows examples of contradiction and entailment relationships based on these features.

**Table 5**
Relationship examples using number, date, and time

| Relationship | Sentence 1 | Sentence 2 |
|---|---|---|
| Entailment | بلغ عدد ضحايا زلزال اليابان 06 قتيلاً<br>The number of Japan earthquake victims reached 60 | زلزال في اليابان وما يزيد عن 05 قتيلاً<br>Earthquake in Japan and more than 50 killed |
| Entailment | مقتل 3 أطفال و 5 نساء في زلزال اليابان<br>3 children and 5 women killed in Japan earthquake | مقتل 8 أشخاص في زلزال اليابان<br>8 people killed in Japan earthquake |
| Contradiction | ولد خالد عام 7891<br>Khaled was born in 1987 | ولد خالد عام 0991<br>Khaled was born in 1990 |
| Contradiction | بلغ عدد ضحايا زلزال في اليابان 06 قتيلًا<br>The number of Japan earthquake victims reached 60 | زلزال في اليابان وما يقل عن 05 قتيل<br>Earthquake in Japan and less than 50 were killed |

In our system, we create a vector to encode each of these regular expressions types (number, time, date) into two values to consider two different cases:

- if quantity value of regular expression is NOT same in both sentences;
- if same quantity value of regular expression is in both sentences.

### 3.3.2. Language models

In this work, we used different language models to represent the pairs of sentences. We compared the results of the following language models:

- **Bag of Words:** A bag of words means an unordered set of words, ignoring their exact positions. The simplest bag-of-words approach represents the context of a target word by a vector of features, each binary feature indicating whether a vocabulary word w does or does not occur in the context. Bag-of-word features are effective at capturing the general topic of the discourse in which the target word has occurred. This in turn tends to identify the senses of a word that are specific to certain domains [22]. In this work, we extracted a bag of words based on words vs. chars in each sentence of the pairs.
- **N-grams:** An n-gram is a continuous sequence of n items from a given sequence of text or speech data. N-gram models assign a conditional probability to possible next words or assign a joint probability to an entire sentence. N-grams are essential in any task in which we have to identify words in noisy ambiguous input [22]. In this work, we extracted unigrams, bigrams, and trigrams for words vs. chars in each sentence of the pairs.
- **TF-IDF** (term frequency–inverse document frequency): This is a term-weighting scheme that is commonly used to represent textual documents as vectors (for purposes of classification, clustering, visualization, retrieval, etc.). Let T = t1,..., tn be the set of all terms that occur in a document corpus under consideration. Then, a document di is represented by an n-dimensional real-valued vector xi = (xi1,..., xin) with one component for each possible term from T. The most common TF–IDF weighting is defined by $x_{ij} = TF_iIDF_j(\sum_{j=0}^{n} j(TF_{ij}IDF_j)^2)^{-}1/2$ [43]. In this work, we extracted TF-IDF based on words vs. chars in each sentence of the pairs.
- **Word Embedding:** This is a low-dimensional word vector that encodes the semantic meanings of words [31]. In this work, we created word2vec models using Genism implementation. The training was done by using 50% of the translated sentences from the SICK and PHEME data sets.

### 3.4. Classification models

In order to detect the relationship type (contradiction, entailment, or neutral) between two sentences, we used different traditional machine-learning classifiers and compared their results. The algorithms that were used were support vector machine (SVM) [47], stochastic gradient descent (SGD) [44], decision tree (DT) [15], ADABoost [1], k-nearest neighbors (KNN) [24], and random forest [38].

## 4. Evaluation & results

We evaluated our proposed solution on our created data set (ArNLI) and on both of the Ar_SICK and Ar_PHEME data sets. Each data set was divided into training and testing sets as 80 and 20%, respectively. Table 6 presents the results of applying the different algorithms on the ArNLI data set.

**Table 6**
Results of experiments on ArNLI

| | | | SVM | SGD | DT | ADA | KNN | RF |
|---|---|---|---|---|---|---|---|---|
| TFIDF | | Char | 0.65 | 0.65 | 0.59 | 0.52 | 0.52 | 0.73 |
| | | Word | 0.63 | 0.65 | 0.57 | 0.51 | 0.57 | 0.7 |
| | | Union | 0.65 | 0.63 | 0.59 | 0.52 | 0.56 | 0.73 |
| **Bag of Words** | | Chars | 0.64 | 0.57 | 0.59 | 0.53 | 0.56 | 0.75 |
| | | Words | 0.61 | 0.65 | 0.57 | 0.55 | 0.6 | 0.71 |
| N-Grams | Words | Unigram | 0.62 | 0.61 | 0.57 | 0.54 | 0.51 | 0.71 |
| | | Bigram | 0.59 | 0.62 | 0.59 | 0.54 | 0.51 | 0.72 |
| | | Trigram | 0.58 | 0.52 | 0.59 | 0.55 | 0.57 | 0.75 |
| | Chars | Unigram | 0.62 | 0.63 | 0.57 | 0.54 | 0.52 | 0.72 |
| | | Bigram | 0.62 | 0.62 | 0.57 | 0.54 | 0.54 | 0.65 |
| | | Trigram | 0.6 | 0.62 | 0.57 | 0.52 | 0.54 | 0.61 |
| W2Vec | | word2vec | 0.57 | 0.59 | 0.57 | 0.52 | 0.53 | 0.67 |
| | | word2vec TF-IDF | 0.57 | 0.55 | 0.56 | 0.55 | 0.59 | 0.66 |

The experiments showed that random forest achieved the best results on the ArNLI data sets (with an accuracy of 0.75). As for the language models that were used in the feature extraction, we found that the best results were achieved by combining the trigrams of word vectors with the contradiction vector or combining the bag of words of the chars vector with the contradiction vector. We applied the different experiments on the automatically translated Ar_PHEME and Ar_SICK data sets. Tables 7 and 8 show the respective accuracy results that were achieved by our experiments on both data sets.

**Table 7**
Result accuracy on AR_PHEME data set

| | | | SVM | SGD | DT | ADA | KNN | RF |
|---|---|---|---|---|---|---|---|---|
| TFIDF | | Char | 0.91 | 0.84 | 0.58 | 0.77 | 0.93 | 1 |
| | | Word | 0.89 | 0.85 | 0.52 | 0.78 | 0.92 | 1 |
| | | Union | 0.89 | 0.84 | 0.58 | 0.77 | 0.93 | 1 |
| Bag of Words | | Chars | 0.94 | 0.88 | 0.57 | 0.78 | 0.91 | 1 |
| | | Words | 0.91 | 0.87 | 0.52 | 0.78 | 0.93 | 1 |
| N-Grams | Words | Unigram | 0.63 | 0.6 | 0.53 | 0.64 | 0.88 | 1 |
| | | Bigram | 0.9 | 0.87 | 0.57 | 0.76 | 0.92 | 1 |
| | | Trigram | 0.92 | 0.88 | 0.52 | 0.76 | 0.91 | 1 |
| | Chars | Unigram | 0.63 | 0.58 | 0.53 | 0.64 | 0.88 | 1 |
| | | Bigram | 0.9 | 0.87 | 0.57 | 0.76 | 0.92 | 1 |
| | | Trigram | 0.92 | 0.89 | 0.52 | 0.76 | 0.91 | 1 |
| W2Vec | | word2vec | 0.47 | 0.43 | 0.46 | 0.56 | 0.8 | 1 |
| | | word2vec TF-IDF | 0.49 | 0.46 | 0.5 | 0.58 | 0.8 | 0.99 |

**Table 8**
Result accuracy on AR_SICK data set

| | | | SVM | SGD | DT | ADA | KNN | RF |
|---|---|---|---|---|---|---|---|---|
| TFIDF | | Char | 0.58 | 0.56 | 0.58 | 0.53 | 0.59 | 0.53 |
| | | Word | 0.52 | 0.48 | 0.64 | 0.55 | 0.58 | 0.52 |
| | | Union | 0.52 | 0.54 | 0.63 | 0.53 | 0.58 | 0.56 |
| Bag of Words | | Chars | 0.58 | 0.6 | 0.6 | 0.66 | 0.52 | 0.57 |
| | | Words | 0.54 | 0.57 | 0.6 | 0.52 | 0.52 | 0.48 |
| N-Grams | Words | Unigram | 0.58 | 0.57 | 0.6 | 0.56 | 0.52 | 0.54 |
| | | Bigram | 0.52 | 0.57 | 0.64 | 0.55 | 0.54 | 0.57 |
| | | Trigram | 0.52 | 0.58 | 0.63 | 0.63 | 0.58 | 0.57 |
| | Chars | Unigram | 0.58 | 0.6 | 0.55 | 0.65 | 0.52 | 0.58 |
| | | Bigram | 0.57 | 0.57 | 0.56 | 0.66 | 0.58 | 0.6 |
| | | Trigram | 0.5 | 0.58 | 0.55 | 0.63 | 0.58 | 0.6 |
| W2Vec | | word2vec | 0.53 | 0.6 | 0.53 | 0.66 | 0.57 | 0.57 |
| | | word2vec TF-IDF | 0.52 | 0.57 | 0.52 | 0.66 | 0.5 | 0.58 |

In Table 6, we notice that the best results (100% accuracy) were achieved when using random forest on the translated Ar_PHEME data set. This can be justified by the fact that the PHEME data set had many repetitions and that the PHEME sentences were initially news headlines that were lexically contradicting (such as "ten people are dead in Airbus crash" and "no one died in Airbus crash") and, thus, can be easily detected.

When applying the different experiments on the automatically translated Ar_-SICK data set, we noticed that the best results (an accuracy of 66%) were achieved when using the ADA algorithm with both W2Vec or bigram on the char level language model (see Table 8).

When comparing the results of the different data sets, we noticed that the worst results were achieved on the Ar_SICK data set (66%). We can justify this by the fact that this data set contained many pairs with semantic abstraction levels and the automatic translation step changed the semantics of one or both sentences (making the original label invalid). Table 9 presents a few examples of entailments that our system failed to detect in the Ar_SICK data set.

Average results were achieved on our data set (ArNLI) (75%), as it included different types of contractions with different levels of semantics. Figures 2, 3, and 4 show comparisons of the results of all of the algorithms on the translations of the PHEME, SemEval2014Task1, and ArNLI data sets, respectively. Figures 5, 6, 7, and 8 show comparisons of the best results on the three data sets that were used (translations of PHEME, SemEval2014Task1, and ArNLI data sets) using support vector machine (SVM), ADABoost, stochastic gradient descent (SGD), and random forest, respectively.

**Table 9**
Examples of translations spoiling semantics

| Sentence 1 | Sentence 2 | Automatic Translation of Sentence 1 | Automatic Translation of Sentence 2 | Original Label |
|---|---|---|---|---|
| A dog is rolling on the ground | A dog is sleeping on the ground | كلب هو المتداول على الأرض (بدلاً من كلب يتدحرج على الأرض) | كلب نائم على الأرض | NEUTRAL |
| A horse is being ridden by a person | A person is riding a horse | ويجري تعصف بها حصان من قبل شخص (بدلاً من حصان يركبه شخص) | شخص يركب الخيل | ENTAILMENT |
| A person is tearing sheets | A man is cutting paper | يكون الشخص ورقة تمزيق (بدلاً من يقوم شخص بتمزيق الملاءات) | رجل يقطع ورقة | NEUTRAL |
| There is no woman cutting broccoli | A woman is cutting broccoli | لا يوجد البروكلي قطع امرأة (بدلاً من لا توجد امرأة تقطع البروكلي) | امرأة تقطع البروكلي | CONTRADICTION |

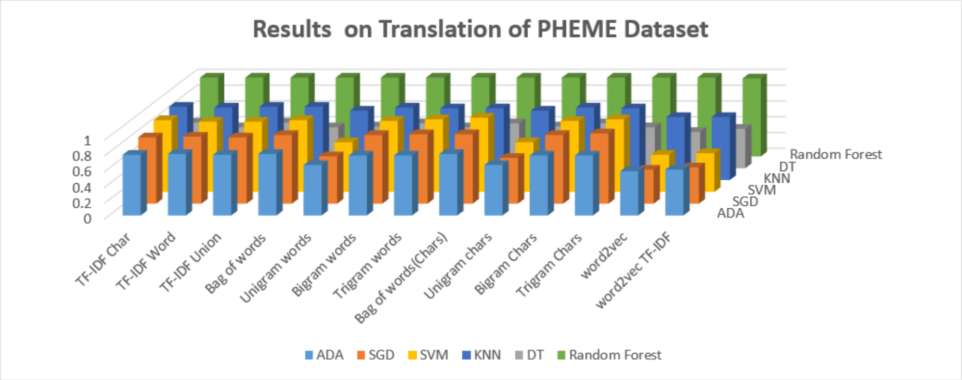

**Figure 2.** Results on translation of PHEME data set

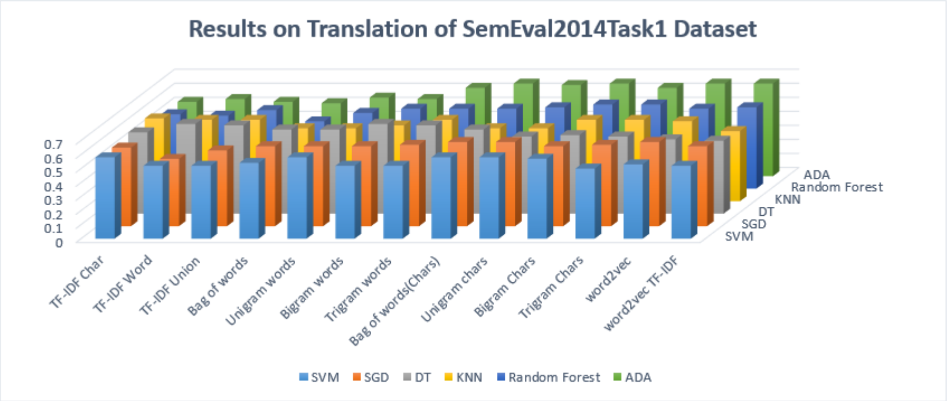

**Figure 3.** Results on translation of SemEval2014Task1 data set

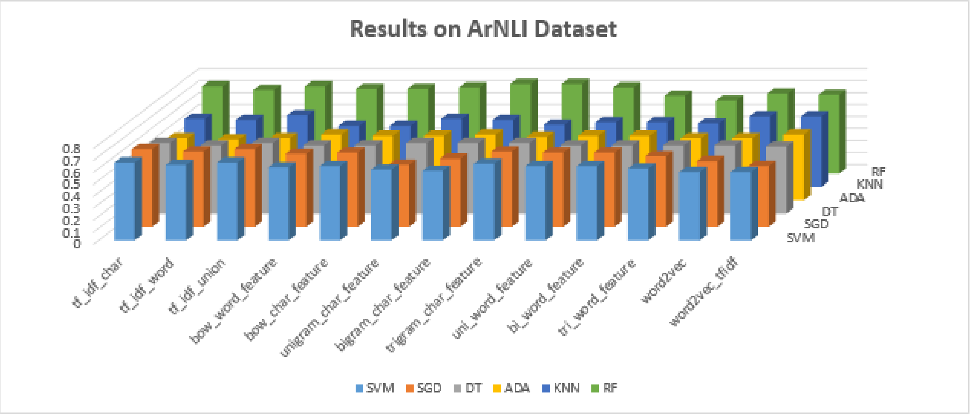

**Figure 4.** Results on ArNLI data set

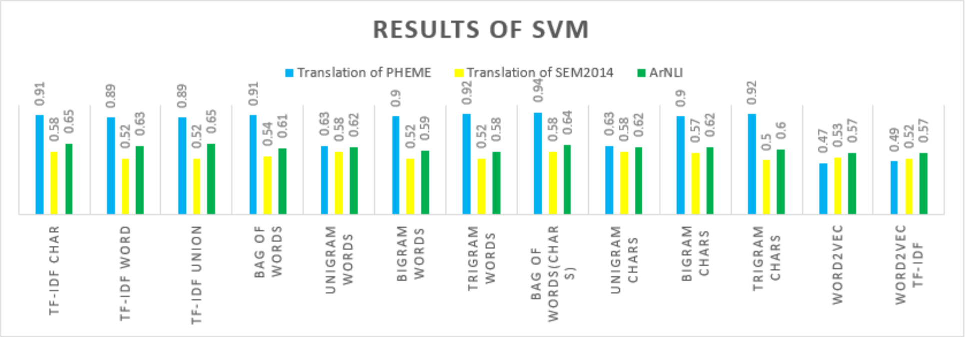

**Figure 5.** Results of SVM

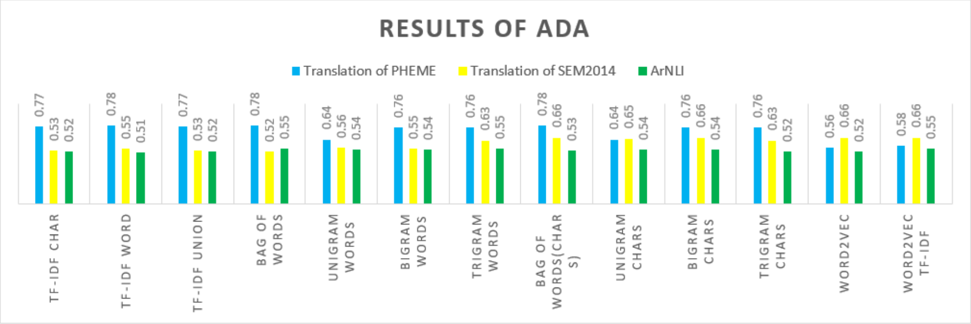

**Figure 6.** Results of ADA

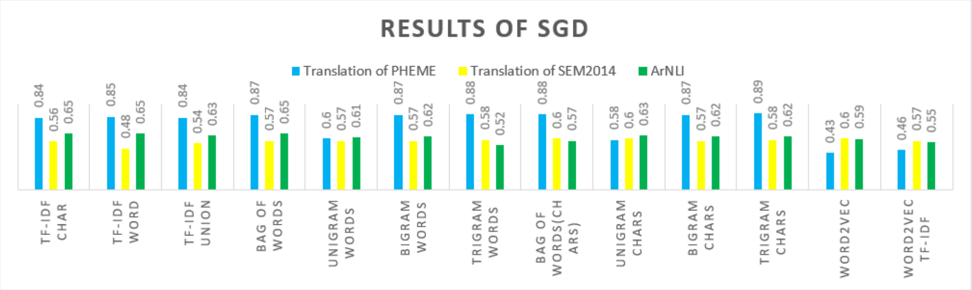

**Figure 7.** Results of SGD

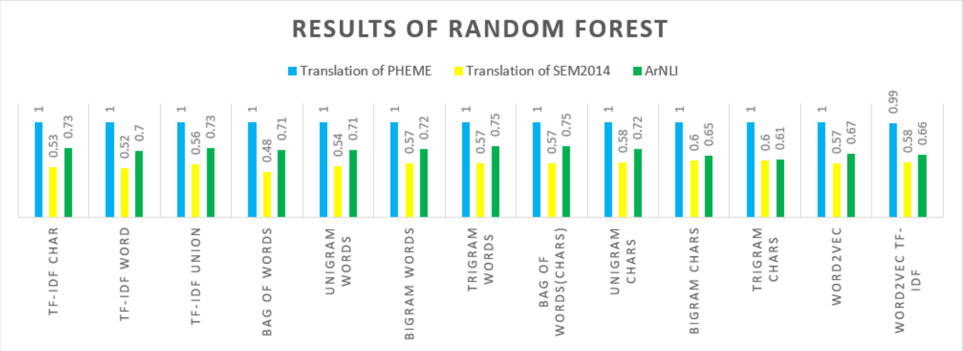

**Figure 8.** Results of random forest

## 5. Conclusion

Detecting entailment relationships between statements is a quite essential and challenging NLP task – especially contradiction detection, which can really optimize the core of many NLP applications. The Arabic language suffers from low resources in NLI detection; only a small data set is available, so no deep-learning solutions have been previously proposed in this domain. In this paper, we presented our semi-automatically created ArNLI data set that contained more than 12k sentences. We automatically translated the English PHEME and SICK data sets. We made some basic experiments to detect entailments in the Arabic language (inspired by Stanford's proposed solutions on the English language). We applied these experiments on our created ArNLI data set and compared the results with translated PHEME and SICK (as to the lack of benchmarks in the Arabic language). The best results of accuracy on the ArNLI data set (0.75) were achieved when using the random forest classifier and a feature vector that contained a combination of the trigram of words vector with the contradiction vector or a combination of the bag of words of chars vector with the contradiction vector.

In a future step, we intend to augment our data set and perform different experiments using different embeddings, different transformers, and different deep-learning algorithms. Moreover, we would like to apply NLI as parts of other important NLP tasks such as sarcasm detection and machine reading.

## References

[1] AdaBoostClassifier scikit-learn.org, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html. Accessed 11 11 2020.

[2] AL-Khawaldeh F.T.: A Study of the Effect of Resolving Negation and Sentiment Analysis in Recognizing Text Entailment for Arabic, 2019. doi: 10.48550/ARXIV.1907.03871.

[3] Alabbas M.: A Dataset for Arabic Textual Entailment. In: *Proceedings of the Student Research Workshop associated with RANLP 2013*, pp. 7–13, INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria, 2013. https://aclanthology.org/R13-2002.

[4] Alabbas M., Ramsay A.: Natural Language Inference for Arabic Using Extended Tree Edit Distance with Subtrees, *J Artif Intell Res*, vol. 48, pp. 1–22, 2013. doi: 10.1613/jair.3892.

[5] Almarwani N., Diab M.: Arabic Textual Entailment with Word Embeddings. In: *Proceedings of the Third Arabic Natural Language Processing Workshop*, pp. 185–190, Association for Computational Linguistics, Valencia, Spain, 2017. doi: 10.18653/v1/W17-1322.

[6] Amirkhani H., AzariJafari M., Pourjafari Z., Faridan-Jahromi S., Kouhkan Z., Amirak A.: FarsTail: A Persian Natural Language Inference Dataset, 2020. doi: 10.48550/ARXIV.2009.08820.

[7] ANli allenai, 2019, https://leaderboard.allenai.org/anli/submissions/get-started.. Accessed 2021.

[8] Ben-Sghaier M., Bakari W., Neji M.: Classification and Analysis of Arabic Natural Language Inference Systems, *Procedia Computer Science*, vol. 176, pp. 551–560, 2020. doi: https://doi.org/10.1016/j.procs.2020.08.057. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.

[9] Bos J., Zanzotto F.M., Pennacchiotti M.: Textual Entailment at EVALITA 2009. In: *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*, vol. 9, 2009.

[10] Boudaa T., Marouani M.E., Enneya N.: Alignment Based Approach for Arabic Textual Entailment, *Procedia Computer Science*, 2019.

[11] Budur E., Özçelik R., Gungor T., Potts C.: Data and Representation for Turkish Natural Language Inference. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8253–8267, Association for Computational Linguistics, Online, 2020. doi: 10.18653/v1/2020.emnlp-main.662.

[12] Clark P.: Recognizing Textual Entailment, QA4MRE, and Machine Reading. In: P. Forner, J. Karlgren, C. Womser-Hacker (eds.), *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, CEUR Workshop Proceedings, vol. 1178, CEUR-WS.org, 2012. http://ceur-ws.org/Vol-1178/CLEF2012wn-QA4MRE-Clark2012.pdf.

[13] Delmonte R., Bristot A., Boniforti M.A.P., Tonelli S.: Entailment and Anaphora Resolution in RTE3. In: *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, p. 48–53, RTE '07, Association for Computational Linguistics, USA, 2007.

[14] Dolan W.B., Brockett C.: Automatically Constructing a Corpus of Sentential Paraphrases. In: *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. https://aclanthology.org/I05-5002.

[15] DecisionTreeClassifier scikit-learn.org, https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. Accessed 11 11 2020.

[16] Eichler K., Gabryszak A., Neumann G.: An analysis of textual inference in German customer emails. In: **SEMEVAL*, 2014.

[17] Fonseca E., Borges dos Santos L., Criscuolo M., Aluisio S.: Overview of the evaluation of semantic similarity and textual inference, 2016.

[18] Harabagiu S., Hickl A., Lacatusu F.: Negation, Contrast and Contradiction in Text Processing. In: *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, p. 755–762, AAAI'06, AAAI Press, 2006.

[19] Hayashibe Y.: Japanese Realistic Textual Entailment Corpus. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 6827–6834, European Language Resources Association, Marseille, France, 2020. https://aclanthology.org/2020.lrec-1.843.

[20] He P., Liu X., Gao J., Chen W.: DeBERTa: Decoding-enhanced BERT with Disentangled Attention., *CoRR*, vol. abs/2006.03654, 2020. http://dblp.uni-trier.de/db/journals/corr/corr2006.html#abs-2006-03654.

[21] Hu H., Richardson K., Xu L., Li L., Kuebler S., Moss L.S.: OCNLI: Original Chinese Natural Language Inference, 2020. doi: 10.48550/ARXIV.2010.05444.

[22] Jurafsky D., Martin J.H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall PTR, USA, 1st ed., 2000.

[23] Khader M., Awajan A.A., Al-Kouz A.: Textual Entailment for Arabic Language based on Lexical and Semantic Matching, *International Journal of Computing*, vol. 12, pp. 67–74, 2016.

[24] KNN Classifier scikit-learn.org, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. Accessed 11 11 2020.

[25] Koreeda Y., Manning C.: ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1907–1919, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021. doi: 10.18653/v1/2021.findings-emnlp.164.

[26] Lendvai P., Augenstein I., Bontcheva K., Declerck T.: Monolingual Social Media Datasets for Detecting Contradiction and Entailment. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 4602–4605, European Language Resources Association (ELRA), Portorož, Slovenia, 2016. https://aclanthology.org/L16-1729.

[27] Li L., Qin B., Liu T.: Contradiction Detection with Contradiction-Specific Word Embedding, *Algorithms*, vol. 10(2), p. 59, 2017. doi: 10.3390/a10020059.

[28] Lingam V., Bhuria S., Nair M., Gurpreetsingh D., Goyal A., Sureka A.: Deep learning for conflicting statements detection in text, 2018. doi: 10.7287/peerj.preprints.26589v1.

[29] Lippi M., Torroni P.: Argumentation Mining: State of the Art and Emerging Trends, *ACM Trans Internet Technol*, vol. 16(2), 2016. doi: 10.1145/2850417.

[30] Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach, *ArXiv*, vol. abs/1907.11692, 2019.

[31] Liu Z., Lin Y., Sun M.: *Representation Learning and NLP*, pp. 1–11, Springer Singapore, Singapore, 2020. doi: 10.1007/978-981-15-5573-2_1.

[32] MacCartney B., Grenager T., de Marneffe M.C., Cer D., Manning C.D.: Learning to recognize features of valid textual entailments. In: *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pp. 41–48, Association for Computational Linguistics, New York City, USA, 2006. https://aclanthology.org/N06-1006.

[33] Marelli M., Bentivogli L., Baroni M., Bernardi R., Menini S., Zamparelli R.: SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 1–8, Association for Computational Linguistics, Dublin, Ireland, 2014. doi: 10.3115/v1/S14-2001.

[34] de Marneffe M.C., Rafferty A.N., Manning C.D.: Finding Contradictions in Text. In: *Proceedings of ACL-08: HLT*, pp. 1039–1047, Association for Computational Linguistics, Columbus, Ohio, 2008. https://aclanthology.org/P08-1118.

[35] Microsoft Research Paraphrase Corpus Microsoft Research, 2005, https://www.microsoft.com/en-us/download/details.aspx?id=52398.. Accessed 2021.

[36] Mishra A., Patel D., Vijayakumar A., Li X., Kapanipathi P., Talamadupula K.: Reading Comprehension as Natural Language Inference:A Semantic Analysis. In: *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pp. 12–19, Association for Computational Linguistics, Barcelona, Spain (Online), 2020. https://aclanthology.org/2020.starsem-1.2.

[37] Padó S., Galley M., Jurafsky D., Manning C.D.: Robust Machine Translation Evaluation with Entailment Features. In: K. Su, J. Su, J. Wiebe (eds.), *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pp. 297–305, The Association for Computer Linguistics, 2009. https://aclanthology.org/P09-1034/.

[38] RandomForestClassifier scikit-learn.org, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed 11 11 2020.

[39] Ritter A., Soderland S., Downey D., Etzioni O.: It's a Contradiction – no, it's not: A Case Study using Functional Relations. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 11–20, Association for Computational Linguistics, Honolulu, Hawaii, 2008. https://aclanthology.org/D08-1002.

[40] Rocha G., Lopes Cardoso H.: Recognizing Textual Entailment: Challenges in the Portuguese Language, *Information*, vol. 9(4), 2018. doi: 10.3390/info9040076.

[41] Romanov A., Shivade C.: Lessons from Natural Language Inference in the Clinical Domain. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1586–1596, Association for Computational Linguistics, Brussels, Belgium, 2018. doi: 10.18653/v1/D18-1187.

[42] Roth D., Sammons M., Vydiswaran V.: A Framework for Entailed Relation Recognition. In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 57–60, Association for Computational Linguistics, Suntec, Singapore, 2009. https://aclanthology.org/P09-2015.

[43] Sammons M., Vydiswaran V., Roth D.: Recognizing Textual Entailment. In: D.M. Bikel, I. Zitouni (eds.), *Multilingual Natural Language Applications: From Theory to Practice*, chap. 6, pp. 209–258, IBM Press, Pearson plc, 2012.

[44] SGD Classifier scikit-learn.org, https://scikit-learn.org/stable/modules/sgd.html. Accessed 11 11 2020.

[45] Şulea O.M.: Recognizing Textual Entailment in Twitter Using Word Embeddings. In: *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pp. 31–35, Association for Computational Linguistics, Copenhagen, Denmark, 2017. doi: 10.18653/v1/W17-5306.

[46] superGlue NYU; FaceBook; DeepMind;UWNLP, 2019, https://super.gluebenchmark.com/. Accessed 2021.

[47] SVM Classifier scikit-learn.org, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Accessed 11 11 2020.

[48] Wang A., Pruksachatkun Y., Nangia N., Singh A., Michael J., Hill F., Levy O., Bowman S.: SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (eds.), *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

[49] Wang A., Singh A., Michael J., Hill F., Levy O., Bowman S.: GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Association for Computational Linguistics, Brussels, Belgium, 2018. doi: 10.18653/v1/W18-5446.

[50] Wang S., Fang H., Khabsa M., Mao H., Ma H.: Entailment as Few-Shot Learner, 2021. doi: 10.48550/ARXIV.2104.14690.

[51] Williams A., Nangia N., Bowman S.R.: A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference, 2017. doi: 10.48550/ARXIV.1704.05426.

[52] Yates A., Banko M., Broadhead M., Cafarella M., Etzioni O., Soderland S.: TextRunner: Open Information Extraction on the Web. In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 25–26, Association for Computational Linguistics, Rochester, New York, USA, 2007. https://aclanthology.org/N07-4013.

## ABBREVIATIONS

NLI: Natural Language Inference

RTE: Recognize Textual Entailment

## ETHICS APPROVAL AND CONSENT TO PARTICIPATE

All authors give ethics approval and consent to participate in the submission and review process.

## CONSENT FOR PUBLICATION

The authors consent for publication.

## AVAILABILITY OF DATA AND MATERIALS

The data set that was created in this research is available in the following repository https://github.com/Khloud-AL/ArNLI

## COMPETING INTERESTS

The authors declare that they have no competing interests.

## FUNDING

The authors declare that they have no funding.

## Affiliations

**Khloud Al Jallad**
Arab International University, Faculty of Information Technology Engineering, Daraa, Syria, k-aljallad@aiu.edu.sy

**Nada Ghneim**
Arab International University, Faculty of Information Technology Engineering, Daraa, Syria, n-ghneim@aiu.edu.sy

**Received**: 08.07.2021
**Revised**: 19.09.2022
**Accepted**: 27.09.2022