algorithms

MDPI

Article

A Lexicon-Based Framework for Mining and Analysis of Arabic Comparative Sentences

Alaa Hamed, Arabi Keshk and Anas Youssef *

Computer Science Department, Faculty of Computers and Information, Menoufia University, Shebin El Kom 32511, Egypt; ams.sakr@yahoo.com (A.H.); arabikeshk@yahoo.com (A.K.)
* Correspondence: anas.youssef@ci.menofia.edu.eg

Abstract: People tend to share their opinions on social media daily. This text needs to be accurately mined for different purposes like enhancements in services and/or products. Mining and analyzing Arabic text have been a big challenge due to many complications inherited in Arabic language. Although, many research studies have already investigated the Arabic text sentiment analysis problem, this paper investigates the specific research topic that addresses Arabic comparative opinion mining. This research topic is not widely investigated in many research studies. This paper proposes a lexicon-based framework which includes a set of proposed algorithms for the mining and analysis of Arabic comparative sentences. The proposed framework comprises a set of contributions including an Arabic comparative sentence keywords lexicon and a proposed algorithm for the identification of Arabic comparative sentences, followed by a second proposed algorithm for the classification of identified comparative sentences into different types. The framework also comprises a third proposed algorithm that was developed to extract relations between entities in each of the identified comparative sentence types. Finally, two proposed algorithms were developed for the extraction of the preferred entity in each sentence type. The framework was evaluated using three different Arabic language datasets. The evaluation metrics used to obtain the evaluation results include precision, recall, F-score, and accuracy. The average values of the evaluation metrics for the proposed sentences identification algorithm reached 97%. The average evaluation values of the evaluation metrics for the proposed sentence type identification algorithm reached 96%. Finally, the average results showed 97% relation word extraction precision for the proposed relation extraction algorithm.

Keywords: natural language processing; Arabic text mining; comparative opinion; comparative sentence identification; type identification; relation extraction; preferred entity extraction

Academic Editors: Affan Yasin, Javed Ali Khan and Lijie Wen

Received: 21 November 2024
Revised: 28 December 2024
Accepted: 6 January 2025
Published: 13 January 2025

Citation: Hamed, A.; Keshk, A.; Youssef, A. A Lexicon-Based Framework for Mining and Analysis of Arabic Comparative Sentences. *Algorithms* **2025**, *18*, 44. https://doi.org/10.3390/a18010044

1. Introduction

A huge portion of comparative opinions in Arabic language is shared daily in social media. Such opinions need to be analyzed for many reasons like improving products and services. In general, comparative opinions are analyzed first by their identification, then by extracting their types [1,2], followed by extracting their relation elements [2–5] and finally extracting the preferred entity [2,6].

Arabic Language has three dialects, namely Modern Standard Arabic (MSA) [7], Quranic Arabic (QA), and Colloquial Arabic [8]. QA is the type of Arabic in which the Quran, the holy book of Islam, is written. In the sixth century A.D., the language was marginally not the same as the Arabic of today. MSA is the most broadly utilized version of

Arabic today in Arabic speaking nations. MSA is utilized as a portion of every media outlet from television to films, to daily newspapers and radio broadcasts. The vast majority of books are written in MSA in addition to politicians'opinions in debates, alongside speeches. MSA is the Arabic dialect that is utilized in everyday life in Arabic speaking countries. Colloquial Arabic is frequently the spoken language of most Arabs. This type of Arabic is subject to regional varieties that not only exist across nations, but also occur in the same nation. The focus of this paper is on Colloquial Arabic and MSA.

Previous work was proposed in the field of comparative opinion mining and analysis of English [9,10], Korean [11], Chinese [12] and Vietnamese [5] languages. Related work specialized in comparative opinion mining and analysis of Arabic language was proposed in [1,3,4,6–8,13]. The contributions of this paper are outlined as follows.

- A comprehensive lexicon-based framework for the detailed mining and analysis of Arabic comparative sentences is proposed and evaluated.
- An algorithm for the identification of Arabic comparative sentences is proposed. The algorithm is referred is to referred to as the Arabic Comparative Sentence Identification (ACSI) algorithm.
- An algorithm for the identification of Arabic comparative sentences types. The algorithm is referred to as Arabic Comparative Sentence Type Identification (ACSTI) algorithm. All types of Arabic comparative sentences are considered in this algorithm. The identified comparative sentences were classified into four different types, namely non-equal gradable, equative, superlative, and non-gradable.
- An algorithm for the extraction of the relation between the different entities in the Arabic comparative sentence is proposed. The algorithm is referred to as the Relation Extraction from Arabic Comparative Sentence (REACS) algorithm. REACS algorithm considers all elements that form any relation which are a relation word, a comparison feature, a first entity, also referred to as entity 1, and a second entity, also referred to as entity 2.
- Two algorithms for the extraction of the preferred entity are proposed. These algorithms are referred to as Preferred Entity Extraction from Arabic Non-Equal Comparative Sentence (PEEANCS) and Preferred Entity Extraction from Arabic Superlative Comparative Sentence (PEEASCS) algorithms.
- An Arabic comparative keyword lexicon is *specifically* developed to evaluate the proposed algorithms. This lexicon contains 649 Arabic comparative keywords that cover all the Arabic comparative sentence types.

Three different datasets had been used in this work to evaluate the proposed framework. The datasets include a Twitter dataset composed of 10,005 sentences in Egyptian dialect [14], an MSA dataset composed of 100 sentences, and a social media dataset composed of 501 sentences in Egyptian dialect. The MSA and social media datasets were manually developed by the authors for the evaluation of the proposed framework. The evaluation metrics used to evaluate the proposed algorithms include precision, recall, F-score and accuracy. For the ACSI algorithm, the average values of the four evaluation metrics over all datasets were in the range of 92% to 97%. For the ACSTI algorithm, the average precision, recall, F-score and accuracy values of the proposed type identification algorithm over the four types using all datasets were 96%, 91%, 92% and 96%, respectively. For the REACS, PEEANCS and PEEASCS algorithms, the average results over all datasets were 97% precision for relation word extraction, 73% precision for feature extraction, 75% precision for first entity extraction, 82% precision for second entity extraction and 65% precision for preferred entity extraction.

The rest of the paper is organized as follows. Related work is presented and discussed in Section 2. Section 3 presents a detailed description of the components that form the

proposed framework. Section 4 discusses the details of the datasets used to obtain the evaluation results. Section 5 presents the evaluation metrics. Section 6 presents and discusses the evaluation results. Finally, Section 6 presents the conclusions and future work.

2. Related Work

There were many papers that were proposed in the field of mining and analysis of comparative sentences in different languages. For example, a set of recent papers were proposed to address comparative sentences in English [9,10], Korean [11], Chinese [12] and Vietnamese [5] languages. Since the focus of our work is on Arabic language [2], this section will only present the work proposed in this language, as will be discussed later.

El-Halees [1] mined comparative sentences in Arabic text. Firstly, Arabic comparative sentences were identified from non-comparative ones using Part Of Speech (POS) tags [15], in which the f-measure was on average 63.73% with low precision. Secondly, three machine learning classifiers were applied, namely K-nearest neighbors (KNN), Support Vector Machine (SVM), and Naive Bayes [16]. The best obtained f-measure was 86.63% using KNN [17]. This work showed that using machine learning is much better than using POS in identifying Arabic comparative sentences. Finally, a combination of SVM and POS was applied which resulted in f-measure of 88.87%. This resulted in little improvement compared to using machine learning only. Moreover, the work in this paper applied another additional task which was the generation of a set of rules to characterize three types of Arabic comparative sentences namely: non-equal gradable, equative and superlative. However, as opposed to our work, a fourth type, namely a non-gradable comparative type, was not investigated in this work.

Alharbi, and Khan [3] used decision tree classifier C4.5 (J48 implementation) [18] to identify Arabic comparative opinions. Although the obtained results were promising, the combination of J48 implementation, keywords, and POS improved the performance more than the performance obtained using only J48 implementation. The keyword classifier was found to be the best technique for detecting gradable comparisons. However, the combination between the three different approaches achieved a good performance and balance between the gradable and non-gradable comparative types.

Eldefrawi et al. [4] proposed work in comparative relation extraction for Arabic language. In this work, the authors addressed comparative relations in MSA, Egyptian and Khaliji Arabic dialects. The Conditional Random Field (CRF) Algorithm [19,20] was used to extract comparative relations and it achieved high accuracy results in extracting the two entities which are compared against each other in any comparative sentence.

Eldefrawi et al. [6] proposed a machine learning technique to identify preferred entities in Arabic comparative opinions. Five main categories were proposed to classify comparison keywords, in order to facilitate the analysis of comparative sentences. The obtained results of identification were an average f-measure of 96.5%.

Alotaibi Najm et al. [13] proposed a deep neural-network based model for the identification of comparative sentences from Arabic social media text. This work implemented the proposed model into three steps. Firstly, the data were processed to be transformed into a useful format. Secondly, the pre-processed data were fed to the model for classification. Finally, a parameter optimizer algorithm was employed for fine tuning the parameters involved in the model to enhance results. The proposed model was evaluated by two standard datasets namely Coprus and Corpus+. The obtained results showed high accuracy, precision, recall and F-score which ranged between 94% and 98%.

The proposed framework is based on an Arabic comparison keyword lexicon, which was manually built for the purpose of operating and evaluating this framework. The evaluation results showed that high accuracy was obtained in all Arabic comparative opinion

analysis steps. To the best of the authors' knowledge, this work presents the first comprehensive research work which implements all steps of the Arabic comparative sentences mining and analysis in which all types of Arabic comparative sentences are investigated.

Furthermore, this paper is the first paper to investigate the Arabic non-gradable sentence type in all framework steps. Table 1 summarizes the differences between the related work and the proposed work. The table shows the main approach applied in each of the related papers together with the objectives of each paper when compared with the proposed work. It is clear from the the table that the proposed work is the only one that comprehensively addressed all steps involved in mining and analysis of Arabic comparative sentences.

Table 1. Summary of related works.

Reference	Approach Type	Comparative Sentence Identification	Comparative Sentence Type Identification	Relation Extraction	Preferred Entity Extraction
El-Halees [1]	Machine Learning	✓	x	x	x
Alharbi and Khan [3]	Lexicon-Based	✓	x	x	x
Eldefrawi et al. [4]	Machine Learning	x	x	✓	x
Eldefrawi et al. [6]	Machine Learning	x	x	x	✓
Alotaibi et al. [13]	Deep Learning	✓	x	x	x
Proposed Work	Lexicon-Based	✓	✓	✓	✓

3. Proposed Framework

This section discusses the detailed steps of the proposed approach framework. The steps of the proposed framework are shown in Figure 1. The steps shown in the figure starts with manually building an Arabic comparative keywords lexicon. Afterwards, data are collected for preprocessing. Identification of each of comparative sentences and comparative sentence type is then applied. This is followed by the extraction of each of the comparative sentence relation and the comparative sentence preferred entity. Finally, the accuracies of identification and extraction are calculated. It should be noted that the proposed framework was developed on a machine with an Intel Core i7-6700, 3.4 GHz processor and a 16 GB RAM. Visual Basic.NET programming language and Visual Studio IDE were used to develop the software code of the framework. The rest of this section presents a detailed explanation of the proposed algorithms used to implement the different components of the framework.

3.1. Arabic Comparative Keywords Lexicon

In this paper, the proposed approach is based on a manually developed Arabic comparison keywords lexicon. This lexicon was manually developed because, to the best of the authors' knowledge, there is currently no standard lexicon that *specifically* contains Arabic comparison keywords. The lexicon was first developed by analyzing the linguistic rules that govern the formation of the Arabic comparative sentence. These rules resulted in the categorization of the Arabic comparative sentences into four different types [1] as will be described in detail later. Secondly, for each sentence type, samples of the comparison keywords and the associated comparative sentences were collected from different web sites including Facebook, Twitter, YouTube comments, and web sites which compare between different products and/or services.

Table 2 lists the comparison keywords used to build the lexicon where the keywords are categorized by their sentence types. The authors would like to note that the developed lexicon does not provide a comprehensive listing of all comparative keywords that exist

in the Arabic language. Therefore, this will affect the accuracy of the next steps of the proposed framework if comparative sentences with keywords that do not exist in the lexicon are analyzed.

Arabic comparative sentences are categorized into four different types as follows [1]:

1. **Non-equal gradable** relation expresses a greater than or less than relation in which an ordering of two entities with respect to some of their features is applied. An example Arabic sentence of such a type is دراسة أوراكل أعمق من مايكروسوفت (studying Oracle is deeper than Microsoft).
2. **Equative** relation expresses a relation which states two objects are equal with respect to some of their features. For an example, an Arabic sentence of such a type is الجامعتان نفس المستوى في التعلم (the two universities have the same level of education).
3. **Superlative** relation expresses a relation that is greater than or less than *all others* or in other words it ranks one object over all others. In Arabic language such a relation adds ال (the) to the comparison word as in االنادى الاهلى المصري الافضل في التاريخ (the Egyptian Club Al-Ahly is the best in the history).
4. **Non-gradable** relation expresses sentences that compare features of two or more objects, but do not grade them. An example Arabic sentence of such a type is تدريس الدكتور رشدي مختلف عن تدريس الدكتور عسى (the teaching style of doctor Roshdi differs from from the teaching style of doctor Esaa).

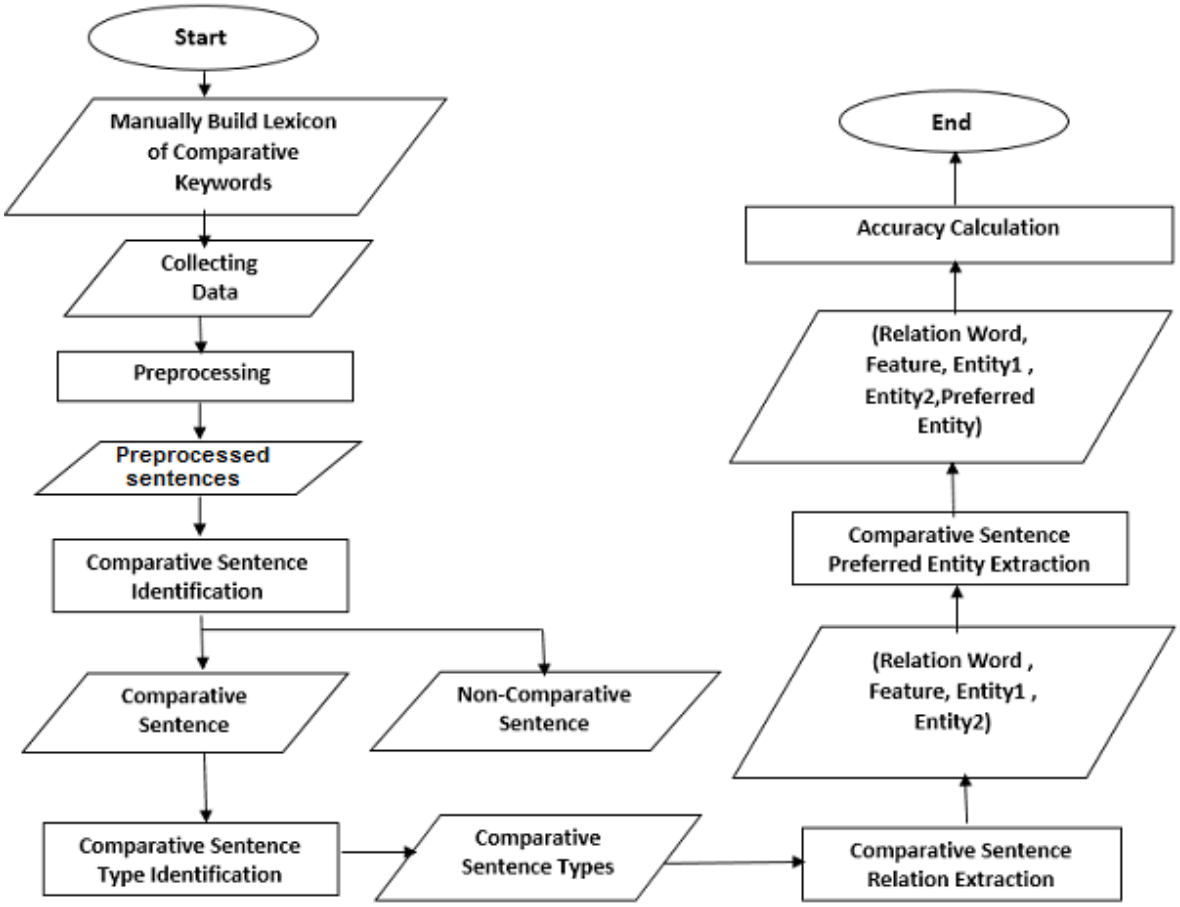

Figure 1. Proposed framework.

Table 2. List of Arabic comparison keywords used to build the lexicon.

Sentence Type	No. of Keywords	Keywords Considered
Non-equal gradable (positive sentiment)	230	أفضل أعمق أجمل أكثر أحسن أعرق أسرع أنعم أسعد أجدى أصدق أبقى أعدل أنور أحلى أكفأ أمجد أشرف أخلص أرق أصفى أقرأ أقرب أشجع أمتع أكتم أحفظ أودع أصبى أعرض أوسع أقبل أبر أرحم أقصى أنصف أطوع أجمد أخضع أزهد أكف أسهل أحب أسمع أفصح أقسط أقوم أمثل أصلح أزيد أولى ألطف أنفع أهدى أخف أثبت أول أمنع أقوى أميز أزهى أشبه أعسل أكرم أعلم أرفع أثمن أعظم أحق أكمل أقدم أحدث أعلى أكبر أغلى أطول أقصر أوضح ألين أقول أهيم أسير أذكى أسلم أعرف أسمى أروع أصح أجمع أظهر أصبر أهدء أحد أرجل أرشد أغنى أوفى أفتح أعز أطهر أحصى أحرص أنظف أبيض أخير أزهر أذخر أطيب أغور أسبق افضل اعمق اجمل اكثر احسن اعرق اسرع انعم اسعد اجدى اصدق ابقى اعدل انور احلى اكفأ امجد اشرف اخلص ارق اصفى اقرأ اقرب اشجع امتع اكتم احفظ اودع اصبى اعرض اوسع اقبل ابر ارحم اقصى انصف اطوع اجمد ازهد اكف اسهل احب اسمع افصح اقسط اقوم امثل اصلح ازيد اولى الطف انفع اهدى اخف اثبت اول امنع اقوى اميز ازهى اشبه اعسل اكرم اعلم ارفع اثمن اعظم احق اكمل اقدم احدث اعلى اكبر اغلى اطول اقصر اوضح الين اقول اهيم اسير اذكى اسلم اعرف اسمى اروع اصح اجمع اصبر اعجل اهدء احد ارجل ارشد اغنى اوفى افتح اعز اطهر احصى احرص ابيض اخير ازهر اذخر اطيب اغور اسبق أشيك اشيك انضف أنضف امرح أمرح أعجب اعجب أشهر اشهر أجدع اجدع
Non-equal gradable (negative sentiment)	67	أقل أسوء أبطأ أخشن أبئس أظلم أسفل أقدر أبعد أبرد أحر أضيق أجمد أقسى أعصى أجمد أغمض أصعب أكفر أنقص أجهل أضر أثقل أخر أضعف أشذ أهزل أهبل أهون أمر أحمض أسود أموت أسمن أحقر أقبح أصغر أرخص أقصر أشد أبيع أغبى أبشع أدنى أشتى أشقى أبخل أفقر أصقع أذل أنكر أنجس أضل أقذر أشر أجشع اقل اسوء ابطأ اخشن ابئس اظلم اسفل اقدر ابعد ابرد احر
Equative	6	نفس ،متساوى ، متساويين، متساوين ،مطابق، نفسه
Superlative (positive sentiment)	236	الأفضل الافضل الافضلان الأفضلان الأفضلون الافضلون الفضلى الفضليات الفضليان الأعمق الاعمق الأجمل الاجمل الأكثر الاكثر الأحسن الاحسن الأعرق الاعرق الأسرع الاسرع الأنعم الانعم الأسعد الاسعد الأجدى الاجدى الأصدق الاصدق الأبقى الابقى الأعدل الاعدل الأنور الانور الأحلى الاحلى الأكفأ الاكفأ الأمجد الامجد الأشرف الاشرف الأخلص الاخلص الأرق الارق الأصفى الاصفى الأقرب الاقرب الأشجع الاشجع الأدرى الادرى الأعلى الاعلى الأكبر الاكبر الأقصى الاقصى الأهم الاهم الأمتع الأمتع الأكتم الاكتم الأحفظ الاحفظ الأودع الاودع الأصبى الأعرض الأوسع الأبر الأرحم الأنصف الأطوع الأجمد الأخضع الأزهد الأكف الأغمض الأسهل الأحب الأفصح الأقسط الأمثل الأصلح الأزيد الأولى الألطف الأنفع الأهدى الأثقل الأول الأمنع الأقوى الاميز الازهى الاشبه الاعسل الاعوم الاكرم الاعلم الارفع الاثمن الاعظم الاحق الاكمل الاقدم الاحدث الاغلى الاطول الاوضح الاشد الالين الاقول الاهيم الاذكى الاسمى الاروع الاصح الاشتى الاظهر الاصبر الاهدء الارجل الارشد الاغنى الاوفى الافتح الاعز الاطهر الاحصى الاحرص الانظف الازهر الاذخر الاطيب الاغور الاسبق الاصبى الاعرض الاوسع الابر الارحم الانصف الاطوع الاجمد الاخضع الازهد الاكف الاغمض الاسهل الاحب الافصح الاقسط الامثل الاصلح الازيد الاولى الالطف الانفع الاهدى الاثقل الاول الاخر الامنع الاقوى الأميز الأزهى الأشبه الأعسل الأعوم الأكرم الأعلم الأرفع الأثمن الأعظم الأحق الأكمل الأقدم الأحدث الأغلى الأطول الأوضح الأشد الألين الأقول الأهيم الأذكى الأسمى الأروع الأصح الأشتى الأظهر الأصبر الأهدء الأرجل الأرشد الأغنى الأوفى الأفتح الأغمق الأعز الأطهر الأحصى الأحرص الأنظف الأزهر الأذخر الأطيب الأغور الأسبق الآخر الاشيك الاشيك الامرح الأمرح الأضخم الاضخم الاعجب الأعجب الأكثر الاكثر الاكتر الأكتر الاشهر الأشهر الأجدع الاجدع الاعجب الأعجب الانضف الأنضف

Table 2. *Cont.*

Sentence Type	No. of Keywords	Keywords Considered
Superlative (negative sentiment)	113	الأقل الاقل الأسوء الاسوء الأبطأ الابطأ الأخشن الاخشن الأبئس الابئس الأظلم الاظلم الأبعد الابعد الأبرد الابرد الأحر الاحر الأضيق الأجهد الأقسى الأعصى الأصعب الأكفر الأنقص الأجهل الأضر الأخف الأخر الأضعف الاشذ الاهزل الاخسر الاهبل الاهون الامر الاحمض الاسمن الاحقر الاقبح الأصغر الارخص الاقصر الاشد الابيع الاغبى الابشع الاعجل الاشقى الابخل الافقر الاصقع الاغمق الاذل الانكر الانجس الاضل الاخفى الاقذر الاثر الاجشع الاضيق الاجهد الاقسى الاعصى الاصعب الاكفر الانقص الاجهل الاضر الاخف الاخر الاضعف الأشذ الأهزل الأخسر الأهبل الأهون الأمر الأحمض الأسمن الأحقر الأقبح الأصغر الأرخص الأقصر الأبيع الأغبى الأبشع الأعجل الأشقى الأبخل الأفقر الأصقع الأغمق الأذل الأنكر الأنجس الأضل الأخفى الأقذر الأثر الأجشع الأصغر الآخر الاهطل الأهطل الاخطر الأخطر الاسوء الأسوء الاسوأ الأسوأ
Non-gradable	3	يختلف، أما، اما
Total	**649**	

3.2. Data Collection

The datasets used to obtain the experimental results in this paper were collected manually from different social web sites, as described before. This section discusses the datasets which were used to evaluate the proposed framework. In this work, three datasets were used including an Arabic Sentiment Tweets Dataset (ASTD) [14], an MSA dataset and a social media dataset. In our work, we refer to the ASTD as the Twitter dataset. The MSA and the social media datasets were manually developed for the evaluation of the proposed framework. Tables 3 and 4 show a brief summary of the three datasets.

It should be noted in Table 4 that some sentences were not be able to be automatically classified in one of the four sentence types. This is due to the following limitations. Firstly, the Twitter dataset was initially used to evaluate the proposed framework when no other datasets were used at that time. This results in a large portion of the sentences not being classified in the Twitter dataset. This limitation did not affect the other two datasets. Secondly, in all used datasets, there were no spaces between some comparison keywords and the words before or after them. This inhibited the correct classification of the sentence type. Finally, missing or not considering the Arabic diacritics drastically affected the correct detection of the comparison keywords.

The authors would like to clarify that they did not adopt the formal machine learning training, validation, and testing methodology, since the proposed framework is only based on the developed comparative keywords lexicon. The steps used to evaluate the different framework algorithms were as follows. The Twitter dataset was initially used for building the initial lexicon. Afterwards, the other two datasets, i.e., MSA and social media, were manually developed and then employed in the evaluation of proposed framework. It should be noted that the Twitter dataset is a standard dataset for Arabic sentiment analysis but is not specifically designed for the mining and analysis of Arabic comparative sentences. Therefore, the expected balance between comparative and non-comparative sentences or the balance among the different sentence types is not guaranteed.

Table 3. Number of comparative and non-comparative sentences in each dataset.

Dataset	Total Number of Sentences	Number of Comparative Sentences	Number of Non-Comparative Sentences
Twitter(ASTD) [14]	10,005	1345	8660
MSA	100	70	30
Social Media	501	217	284

Table 4. Classification of comparative sentence types in each dataset.

Dataset	Number of Non-Equal Gradable Sentences	Number of Equative Sentences	Number of Superlative Sentences	Number of Non-Gradable Sentences	Sentences Not Classified
Twitter	36	36	523	29	721
MSA	24	6	29	3	8
Social Media	20	10	166	20	1

3.3. Data Preprocessing

This section discusses the detailed steps of preprocessing the Arabic sentences that form each dataset. Firstly, each sentence in the dataset is accessed in order. Secondly, each word, i.e., separated by a space, in the accessed sentence is checked for the presence of any characters that do not exist in the set of 28 Arabic alphabets from أ (the alphabetic letter A or a) to ى (the alphabetic letter Z or z) and the set of decimal digits from 0 to 9. Any word that does not satisfy these conditions is completely removed. Therefore, all words that contain any special characters like punctuation, exclamation, question marks, etc., . . . are removed. Finally, the preprocessed Arabic sentence becomes available for the evaluation of further steps in the proposed framework.

3.4. Arabic Comparative Sentence Identification

This section describes the detailed steps of the ACSI algorithm. Identification of Arabic comparative sentences was performed by searching for Arabic comparison keywords in the sentence using the manually built Arabic comparative keywords lexicon. If there is an Arabic comparison keyword in the sentence, then it is a valid Arabic comparative sentence; otherwise, it is not.

Algorithm 1 shows the pseudo-code of the ACSI algorithm. As shown in the pseudo-code, the statement in line 1 accesses each sentence in order, followed by the statement in line 2 which splits each sentence into words using the space between words and stores it in an ordered list. The statement in line 3 accesses each word stored in the ordered list followed by the statement in line 4 which accesses each Arabic comparison keyword from the Arabic comparison keywords lexicon in order. The statements from line 5 to line 9 extract the Arabic comparative sentences where the statement in line 5 searches for an Arabic comparative keyword in the sentence and if the comparative keyword exists, the statement in line 7 records the sentence as an Arabic comparative sentence.

Algorithm 1 Pseudo-code of the ACSI algorithm

Input: dataset of preprocessed sentences.
Output: dataset of comparative sentences.

```
for each sentence_i in the dataset of preprocessed sentences do
    New Array arr[] = split sentence_i with " "
    for each word_j in arr[] do
        for each keyword_n in Comparative Keyword Lexicon do
            if (word_j = keyword_n) then
                sentence_i is identified as a "Comparative Sentence"
                insert sentence_i in dataset of comparative sentences
                break
            end if
        end for
    end for
end for
```

3.5. Arabic Comparative Sentence Type Identification

This section describes the detailed steps of the ACSTI algorithm. In general, comparative sentence are categorized into four types, namely non-equal gradable, equative, superlative and non-gradable comparative [1]. Identifying an Arabic comparative sentence type is based on the type of the Arabic comparative keyword which exists in the sentence. This type is determined based on the Arabic comparative keywords lexicon where every type has its own Arabic comparative keywords. The four types are described as follows, together with some examples for the sentences that represent each type [1]:

1. **Non-Equal Gradable Comparison Type:** Relations of this type express an ordering of objects with regard to some of their features. An example of this sentence type is the sentence that contains the Arabic comparative keyword, whose format is أفعل. This keyword has an original verb that consists of three letters. An example of such sentence is دراسة أوراكل أعمق من مايكروسوفت (studying Oracle is deeper than Microsoft) where the comparative keyword is directly mentioned in the comparative sentence. On the contrary, if the verb contains more than three letters, the sentence will contain the Arabic word اقل او اكثر (less or more) and the sentence will be like the following سعيد أكثر إجتهادا من أخيه (Said has more diligence than his brother).
2. **Equative Comparison Type:** Relations of this type state that two objects are equal with respect to some of their features. An example of such sentence is الجامعتان نفس المستوى فى التعليم (the two universities have the same level of education).
3. **Superlative Comparison Type:** Relations of this type ranks one object over other objects. In Arabic language this type may add ال to the comparison word or not. Examples of this type are الأهلى المصرى الأفضل فى العالم (the Egyptian Club Al-Ahly is the best in the history) and رونالدو أفضل لاعب فى العالم (Ronaldo is the best player in the world).
4. **Non-gradable Comparison Type:** Non-gradable comparative sentences type compares features of two or more objects, but do not grade them. There are three subtypes as follows:
    - Object A is similar to or different from object B with regard to some features. An example of this type is تدريس الدكتور رشدى يختلف عن تدريس الدكتور عيسى (the teaching style of doctor Roshdi differs from from the teaching style of doctor Esaa).

- Object A has a feature f1, and object B has another feature f2 where f1 and f2 can substitute each other. An example of this type is الكمبيوتر المكتبى يستخدم سماعات خارجية أما اللاب توب يستخدم سماعات داخلية (the desktop computer uses external speakers while the laptop uses internal speakers).
- Object A has a certain feature, but object B does not have it. An example of this type is جوال أ يستخدم سماعات أذن وجوال ب لا يستخدم (mobile A uses headphones while mobile B does not).

Algorithm 2 shows the pseudo-code of the ACSTI algorithm. As shown in the pseudo-code, the statement in line 1 accesses each Arabic comparative sentence in order and the statement in line 2 splits an Arabic comparative sentence into words using the space between words and puts each word in an array in order. The statement in line 3 accesses each word in the array in order. The statement in line 4 accesses each comparison keyword from the Arabic comparison keywords lexicon in order. The statements from line 5 to line 11 check if the word, $word_j$ in the Arabic comparative sentence exists in the Arabic non-equal comparison keywords. If the word next to $word_j$ is من (than), then the Arabic sentence is considered an Arabic non-equal gradable comparative sentence. The statements in lines 12 to 14 check if $word_j$ is an Arabic equative comparison keywords then the Arabic sentence is considered an Arabic equative comparative sentence. Similarly, the statements in lines 15 to 20 check if the sentence under consideration is either an Arabic superlative comparative sentence or a non-gradable comparative sentence.

Algorithm 2 Pseudo-code of the ACSTI algorithm

Input: dataset of comparative sentences.
Output: dataset of comparative sentences with identified sentence types

```
1:  for each comparative-sentence_i in the dataset of comparative sentences do
2:      New Array arr[] = split comparative-sentence_i with " "
3:      for each word_j in arr[] do
4:          for each keyword_n in Comparative Keyword Lexicon do
5:              if (word_j = NonEqualTypeKeyword_n) then
6:                  if (j + 1 < arr[].Length − 1) then
7:                      if (word_j+1 = من) then
8:                          set type of comparative-sentence_i to "NonEqual Type"
9:                      end if
10:                 end if
11:             end if
12:             if (word_j = EquativeTypeKeyword_n) then
13:                 set type of comparative-sentence_i to "Equative Type"
14:             end if
15:             if (word_j = SuperlativeTypeKeyword_n) then
16:                 set type of comparative-sentence_i to "Superlative Type"
17:             end if
18:             if (word_j = NonGradableTypeKeyword_n) then
19:                 set type of comparative-sentence_i to "NonGradable Type"
20:             end if
21:         end for
22:     end for
23: end for
```

3.6. Relation Extraction

This section discusses the detailed steps applied in the relation extraction process in each of the four types of Arabic comparative sentences mentioned in the previous section. The relation in any comparative sentence is expressed with the following relation vector (a relation keyword, a feature, the first entity, the second entity, the relation type). For

example, in the comparative sentence "Canon's optics is better than those of Sony and Nikon.", the corresponding relation vector is (better, optics, Canon, (Sony and Nikon), non-equal gradable). Extracting a relation vector from an Arabic comparative sentence depends on the Arabic comparative sentence type. The following are examples of different sentences that illustrate the relation extraction from each of the four Arabic comparative sentences types mentioned in the previous section.

An example of the Arabic non-equal gradable comparative sentence type is بطارية موبايل سامسونج أحسن من بطارية النوكيا (Samsung's mobile battery is better than Nokia's battery) and its extracted relation vector is (أحسن، بطارية، موبايل سامسونج، النوكيا, non-equal gradable) (better, battery, Samsung's mobile, Nokia, non-equal gradable).

An example of the Arabic equative comparative sentence type is الجامعتان نفس المستوى فى التعليم (the two universities have the same level of education) and its extracted relation vector is (نفس، المستوى، الجامعتان, equative) (same, level of education, the two universities, equative). In this sentence, there is no obvious second entity because (الجامعتان) (the two universities) represents both the first and the second entities in the relation vector.

Two examples of the Arabic superlative comparative sentence type are illustrated as follows. The first sentence is الأهلى الأفضل فى التاريخ (Al-Ahly is the best in the history) and its extracted relation vector is (الأفضل، فى التاريخ، الأهلى), superlative) (the best, in the history, Al-Ahly), superlative. The second sentence is أنت خير الوارثين and its extracted relation vector is (خير، الوارثين، أنت), superlative) (the best, heir, You, superlative).

Three examples of the Arabic non-gradable comparative sentence type are illustrated as follows: The first sentence is تدريس الدكتور رشدى يختلف عن تدريس الدكتور عيسى (the teaching style of doctor Roshdi differs from the teaching style of doctor Esaa) and its extracted relation vector is (يختلف عن، تدريس، رشدى الدكتور، الدكتور عيسى), non-gradable) (differs, teaching style, doctor Roshdi, doctor Esaa, non-gradable). The second sentence is الكمبيوتر المكتبى يستخدم سماعات خارجية أما اللاب توب يستخدم سماعات داخلية (the desktop computer uses external speakers while the laptop uses internal speakers) and its extracted relation vector is (أما، سماعات، الكمبيوتر المكتبى، اللاب توب, non-gradable) (while, speakers, desktop computer, laptop, non-gradable). Finally, the third sentence is جوال أ يستخدم سماعات أذن وجوال ب لا يستخدم (mobile A uses headphones while mobile B does not) and its extracted relation vector is (و، سماعات، جوال أ، جوال ب, non-gradable) (while, headphones, mobile A, mobile B, non-gradable).

It should be noted that in the Arabic non-gradable comparative sentence type, each entity 1 and entity 2 can be more than one word, because the two entities are included in the sentence without a standard order. An example of this issue is illustrated in the sentence الكمبيوتر المكتبى يستخدم سماعات خارجية أما اللاب توب يستخدم سماعات داخلية (the desktop computer uses external speakers while the laptop uses internal speakers). In this sentence, entity 1 should be الكمبيوتر المكتبى (desktop computer), entity 2 should be اللاب توب (laptop), and the comparison feature should be يستخدم سماعات خارجية و داخلية (external speakers and internal speakers). Therefore, it cannot be exactly determined how

many words will represent each of entity 1, entity 2 and the comparison feature. To resolve this problem, in our work, we extracted each of entity 1 and entity 2 as the set that is formed of 2–5 words that appear in the non-gradable comparative sentence type before and after the comparison keyword, respectively.

Algorithm 3 shows the pseudo-code of the proposed REACS algorithm specifically for Arabic non-equal comparative sentence type. In the discussion of the REACS algorithm, there will be some notes on the differences between the relation extraction steps that apply for each of the other three Arabic comparative sentence types.

Algorithm 3 Pseudo-code of the REACS algorithm

Input: dataset of NonEqual comparative sentences.
Output: extracted relations from the input dataset.

```
1:  for each sentence_i in the dataset of NonEqual comparative sentences do
2:      New Array arr[] = split sentence_i with " "
3:      for each word_j in arr[] do
4:          for each NonEqual-Type-keyword_n in Comparative Keyword Lexicon do
5:              if word_j = NonEqualTypeKeyword_n then          ▷ Relation Word Extraction
6:                  if (j + 1 > arr[].Length − 1) then
7:                      if (word_{j+1} = من) then
8:                          RelationWord = word_j + " " + word_{j+1}
9:                      end if
10:                 end if                                      ▷ Feature Extraction
11:                 if (j + 4 < arr[].Length − 1) then
12:                     Feature = word_{j+3} + " " + word_{j+4}
13:                 else
14:                     if (j + 3 < arr[].Length − 1) then
15:                         Feature = word_{j+3}
16:                     else
17:                         Feature = " "
18:                     end if
19:                 end if                                      ▷ Entity1 Extraction
20:                 if (j − 2 >= 0) then
21:                     Entity1 = word_{j−1} + " " + word_{j−2}
22:                 else
23:                     if (j − 1 >= 0) then
24:                         Entity1 = word_{j−1}
25:                     else
26:                         Entity1 = " "
27:                     end if
28:                 end if                                      ▷ Entity2 Extraction
29:                 if (j + 3 < arr[].Length − 1) then
30:                     Entity2 = word_{j+3} + " " + word_{j+2}
31:                 else
32:                     if (j + 2<arr[].Length − 1) then
33:                         Entity2 = word_{j+2}
34:                     else
35:                         Entity2 = " "
36:                     end if
37:                 end if
38:             end if
39:         end for
40:     end for
41: end for
```

As shown in Algorithm 3, the statements from line 1 to 5 access each Arabic non-equal comparison keyword from the comparison keywords lexicon in order. The statements

from line 5 to line 10 extract a relation word which is an Arabic non-equal comparison keyword and the next Arabic word من (than). The statements from line 11 to line 19 extract the comparison feature which is the fourth word after the Arabic non-equal comparison keyword such as أحمد أفضل من على فى المذاكرة (Ahmed is better than Ali in studying) so the word المذاكرة (studying) is the comparison feature. The statements from line 20 to line 28 extract entity 1 by searching for the words before the Arabic non-equal comparative type comparison keyword. The statements from line 29 to line 37 extract entity 2 by searching for the words after the Arabic non-equal comparative type comparison keyword.

For the non-equal comparative sentence type, if the detected relation word is أكثر (more) or أقل (less) then in the statements from line 5 to line 10, the extracted relation word will be أكثر (more) or أقل (less), respectively, in addition to the second word, i.e., من (than), after the detected relation word. For example, in this sentence أحمد أكثر اجتهادا من على (Ahmed is more diligent than Ali), the extracted relation word becomes أكثر من (more than).

In the statements from line 11 to line 19, the feature is decided to be the word next to the relation word. In the statements from line 20 to line 28, entity 1 is decided to be the word before the relation word and in the statements from line 29 to line 37, entity 2 is decided to be the third word after the relation word. For example, in this sentence أحمد أكثر اجتهادا من على (Ahmed is more diligent than Ali), the feature becomes اجتهادا (diligence), entity 1 becomes أحمد (Ahmed) and entity 2 becomes على (Ali).

For the other Arabic comparative sentences types, the difference in the REACS algorithm is in the statement in line 1 that accesses each sentence type in order. The statement in line 2 splits each comparative sentence type into words using the space between the words and stores it in an array in order. The statement in line 3 accesses each word in the array in order. The statement in line 4 accesses the comparison keywords that represent each sentence type from the Arabic comparison keywords lexicon. The statements from line 5 to line 10 extract the relation word which corresponds to each type.

For the equative sentence type, the statements from line 11 to line 19 extract the feature which is the next word after the comparison keyword نفس (same). If the comparison keyword is متساويان (equal), then the feature becomes all words starting from the second after the comparison keyword. For example, in the sentence الجامعتان متساويتان فى مستوى التعليم (the two universities have the same level of education), the feature becomes مستوى التعليم (level of education). If the comparison keyword is متساوى (equal) then the feature becomes all words staring from the fourth after the comparison keyword such as على متساوى مع احمد فى الطول والعمر (Ali is equal to Ahmed in length and age), so the feature becomes الطول والعمر (length and age).

For the superlative sentence type, the statements from line 11 to line 19 extract the feature which becomes the next words after the comparison keyword. For example, in the sentence الاهلى الافضل فى التاريخ (Al-Ahly is the best in the history), the feature becomes فى التاريخ (in the history).

For the equative, superlative and non-gradable comparative sentence types, the statements from line 20 to line 28 extract entity 1 by searching for the words before the comparison keyword. The statements from line 25 to line 31 extract entity 2 by searching for the words after the Arabic equative or non-gradable comparative type comparison keyword only. This does not apply for the superlative comparative sentence type, since entity 2 does not exist in this type.

Finally, for Arabic non-equal, equative and superlative comparative sentence types, entity 1 and entity 2 are extracted as two words. However, for the Arabic non-gradable type, entity 1 and entity 2 are extracted as five or six words each because entity 1 and entity 2 cannot be exactly determined. This is because entity 1 and entity 2 are parts of the Arabic non-gradable sentence before and after the relation word such as الكمبيوتر المكتبى يستخدم سماعات خارجية أما اللاب توب يستخدم سماعات داخلية (the desktop computer uses external speakers while the laptop uses internal speakers).

3.7. Preferred Entity Extraction

This section describes the preferred entity extraction from Arabic comparative sentences. There are only two comparative sentences types which have a preferred entity. These are non-equal and superlative comparative types. The preferred entity extraction proposed approach extracts the preferred entity depending on the sentiment of the comparison keyword. In non-equal comparative sentence type, if the comparison keyword has a positive sentiment then entity 1 is the preferred entity, while if the comparison keyword has a negative sentiment then entity 2 is the preferred entity. In superlative comparative sentence type, there is only one entity. Therefore, in superlative comparative sentence type, if the comparison keyword has a positive sentiment then the entity mentioned in the sentence is the preferred entity, while if the comparison keyword has a negative sentiment, then there is no preferred entity in the sentence.

3.7.1. Non-Equal Gradable Type

This section describes the details of the PEEANCS algorithm. The following are some examples for the extraction of the preferred entity based on the sentiment of the comparison keyword. In the Arabic non-equal comparative sentence type أحمد أفضل من على فى الدراسة (Ahmed is better than Ali in studying), the comparison keyword أفضل (better) has a positive sentiment, so entity 1, i.e., (أحمد) (Ahmed), is the preferred entity. In the example نوكيا أسوء من سامسونج (Nokia is worse than Samsung), the comparison keyword, i.e., (أسوء) (worse), has a negative sentiment. Therefore, entity 2, i.e., (سامسونج) (Samsung), is the preferred entity.

Algorithm 4 shows the pseudo-code of the PEEANCS algorithm. As shown in the pseudo-code, the statement in line 1 accesses each non-equal comparative sentence in order. The statement in line 2 splits each non-equal comparative sentence into words using the space between words and stores the words in an array. The statement in line 3 accesses each word in the array. The statement in line 4 accesses each non-equal type comparison keyword from the comparison keywords stored in the lexicon in order. The statement in line 5 checks if the comparison keyword has a positive sentiment, then the statements from line 6 to line 19 extract entity 1 from the words before the comparison keyword. The statement in line 21 checks if the comparison keyword has a negative sentiment, then the statements from line 20 to line 37 extract entity 2 from the words after the comparison keyword.

Algorithm 4 Pseudo-code of the PEEANCS algorithm

Input: dataset of NonEqual comparative sentences.
Output: extracted preferred entities from the input set.

```
for each sentence_i in the dataset of NonEqual comparative sentences do
    New Array arr[] = split sentence_i with " "
    for each word_j in arr[] do
        for each NonEqualTypeKeyword_n in Comparative Keyword Lexicon do ▷ Positive NonEqual type keyword
            if (word_j = PositiveNonEqualTypeKeyword_n) then
                if (j + 1 > arr[].Length − 1) then
                    if (word_{j+1} = من) then
                        if (j − 2 >= 0) then
                            Entity1 = word_{j−1} + " " + word_{j−2}
                        else
                            if (j − 1 >= 0) then
                                Entity1 = word_{j−1}
                            else
                                Entity1 = " "
                            end if
                        end if
                    end if
                end if
                PreferredEntity = Entity1 ▷ Negative NonEqual type keyword
            else
                if (word_j = NegativeNonEqualTypeKeyword_n) then
                    if (j + 1 > arr[].Length − 1) then
                        if (word_{j+1} = من) then
                            if (j + 3 > arr[].Length − 1) then
                                Entity2 = word_{j+3} + " " + word_{j+2}
                            else
                                if (j + 2 > arr[].Length − 1) then
                                    Entity2 = word_{j+2}
                                else
                                    Entity2 = " "
                                end if
                            end if
                        end if
                    end if
                    PreferredEntity = Entity2
                end if
            end if
        end for
    end for
end for
```

3.7.2. Superlative Type

This section describes the details of the PEEASCS algorithm. The following are some examples for the extraction of the preferred entity based on the positive or negative comparison keyword. Comparative superlative type sentence has only one entity if the comparison keyword has a positive sentiment. An example of such sentence is الأهلى الأفضل فى التاريخ (Al-Ahly is the best in the history) in which entity 1, i.e., الأهلى (Al-Ahly), is the preferred entity since the comparison keyword, i.e., الأفضل (the best), has a positive sentiment. In another sentence like الهلوكوست الأبشع فى التاريخ (Holocaust is the worst in the history), the

comparison keyword, i.e., الأبشع (the worst), has a negative sentiment, therefore, there is no preferred entity at all.

Algorithm 5 shows the pseudo-code of the PEEASCS algorithm. As shown in the pseudo-code, the statement in line 1 accesses each superlative comparative sentence in order. The statement in line 2 splits each sentence into words using the space between words and stores it in an array. The statement in line 3 accesses each word in the array. The statement in line 4 accesses each superlative type comparison keywords from the comparison keywords lexicon in order. The statement in line 5 checks if the comparison keyword has a positive sentiment, then the statements from line 6 to line 15 extract entity 1 from the words before the comparison keyword. The statement in line 17 checks if the comparison keyword has a negative sentiment, then the statement in line 18 stores the setting that shows that there is no preferred entity extracted from the analyzed sentence.

Algorithm 5 Pseudo-code of the PEEASCS algorithm

Input: dataset of Superlative comparative sentences.
Output: extracted preferred entities from the input dataset.

```
1:  for each sentence_i in the dataset of Superlative comparative sentences do
2:      New Array arr[] = split sentence_i with " "
3:      for each word_j in arr[] do
4:          for each Superlative-Type-keyword_n in Comparative Keyword Lexicon do ▷
    Positive Superlative Type Keyword
5:              if (wordj = PositiveSuperlativeTypeKeyword_n) then
6:                  if (j − 2 >= 0) then Then
7:                      Entity1 = word_{j−2} + " " + word_{j−1}
8:                  else
9:                      if (j − 1 >= 0) then
10:                         Entity1 = word_{j−1}
11:                     else
12:                         Entity1 = " "
13:                     end if
14:                 end if
15:                 PreferredEntity = Entity1          ▷ Negative Superlative Type keyword
16:             else
17:                 if (word_j = NegativeSuperlativeTypeKeyword_n) then
18:                     PreferredEntity = "No Preferred Entity"
19:                 end if
20:             end if
21:         end for
22:     end for
23: end for
```

4. Evaluation Metrics

This section presents the evaluation metrics used in evaluating the proposed algorithms. Four main standard evaluation metrics were used to obtain the evaluation results. The metrics are *Precision*, *Recall*, *F-score* and *Accuracy* [21]. The following equations define the evaluation metrics. It should be noted that TP and TN parameters represent the number of true positive and true negative identified results, respectively, while FP and FN parameters represent the number of false positive and false negative identified results, respectively.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

$$F\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (4)$$

5. Results and Discussion

5.1. Evaluation of ACSI Algorithm

This section presents and discusses the evaluation results of the proposed ACSI algorithm. Figure 2 and Table 5 show the evaluation results of the ACSI algorithm for each of the three datasets used in the evaluation. The figure shows the results in a graphical presentation so that the reader can easily compare the different metrics accross all datasets. The table shows the same results; however, in a numerical format so that curious reader can accurately compare the different metrics in a quantitative way. The evaluation metrics were calculated using Equations (1) to (4). The TP and FP parameters mentioned in the equations indicate the number of comparative sentences that were truly and falsely identified using the proposed ACSI algorithm, respectively. The TN and FN parameters indicate the number of non-comparative sentences that were truly and falsely identified using the proposed ACSI algorithm, respectively.

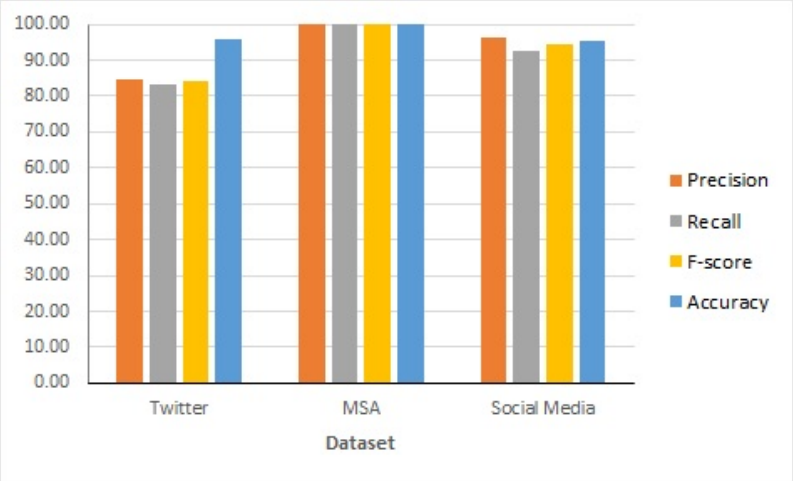

Figure 2. Graphical evaluation results of ACSI algorithm.

Table 5. Numerical evaluation results of ACSI algorithm.

Dataset	Precision	Recall	F-Score	Accuracy
Twitter	84.5	83.49	83.99	95.72
MSA	100	100	100	100
Social Media	96.17	92.63	94.37	95.21

Discussion

This section discusses some limitations of applying the ACSI algorithm. The limitations are described as follows. The listed limitations are left for future improvements on the currently proposed ACSI algorithm.

- The ACSI algorithm only considers the Arabic words in the modern standard Arabic language, which is different in many of its words and meanings from the Egyptian colloquial as well as other Arabic dialects such as Gulf dialect.

- Diacritics in modern standard Arabic language change the meaning of the word completely such as the word نَعم which means yes in English; however, (نِعم) means the best in English. In this example, if the correct diacritics is considered in the analysis of the sentence, the sentence would be identified as a superlative comparative type. The ACSI algorithm does not take into account the diacritics of the words.
- The Arabic word لكن (but or however) may complicate some Arabic sentences and provide them with two different sentiments in the same time. Ana example of this is the Arabic sentence موبايل أ أحسن من ب لكن موبيل ب أحسن فى البطارية (mobile A is better than mobile B, but mobile B has better battery). In this sentence, two comparative sentences, i.e.,موبايل أ أحسن من موبيل ب and موبايل ب أحسن فى البطارية, are combined together. The ACSI algorithm can identify each of these two sentences per se; however, it cannot detect the relation between them.
- Some Arabic comparison keywords do not indicate a comparison at all depending on the context of the sentence. For example in the sentence لا يوجد أى أحد بالمكان (there is nobody in place), the word (أحد) means in English, nobody, not the sharpest like in the sentence (هذا السيف من أحد السيوف) (this sword is one of the sharpest swords). In the later sentence, the word (أحد) (sharpest) is considered a comparison keyword while in the former sentence, it is not. Such limitation also faces the application of the ACSI algorithm.
- Some Arabic comparative keywords cannot detected by ACSI algorithm because of extra characters added to the keyword. For example, the superlative comparative sentence ستم إعطاء جائزة لأفضل طالب فى المدرسة (a prize will be given to the best student in the school), cannot be truly identified using the ACSI algorithm. The reason for this is the presence of the comparison keyword لأفضل (to the best) by adding the Arabic character ل (to) to it. This limitation can be resolved by adding comparison keywords like لأفضل, to the best, to the developed lexicon.
- The exclamation Arabic sentence can be falsely identified as a comparative sentence such as ما أسرع النزول! (how quickly is getting down!). The presence of the exclamation symbol at the end of the sentence is not taken into account when applying the ACSI algorithm.

5.2. Evaluation of ACSTI Algorithm

This section discusses the evaluation results of the proposed ACSTI algorithm. Figure 3 and Table 6 show the evaluation results of the ACSTI algorithm for each comparative sentence type using each of the three considered datasets. The evaluation results of the ACSTI algorithm were calculated using Equations (1) to (4). The TP and FP parameters mentioned in the equations indicate the number of comparative sentences of a certain type that were truly and falsely identified of this type using the proposed ACSTI algorithm, respectively. The TN and FN parameters indicate the number of comparative sentences of a certain type that were truly and falsely identified as a different type using the proposed ACSTI algorithm, respectively. Each comparative sentence type has a separate subfigure shown in Figure 3. As shown in Figure 3a–d, for all sentence types in all datasets, the average precision is 96%, the average recall of the ACSTI algorithm is 91%. the average F-score of the ACSTI algorithm is 92% and the average accuracy is 96%. A detailed discussion of the results shown in this figure will be presented in the next subsections.

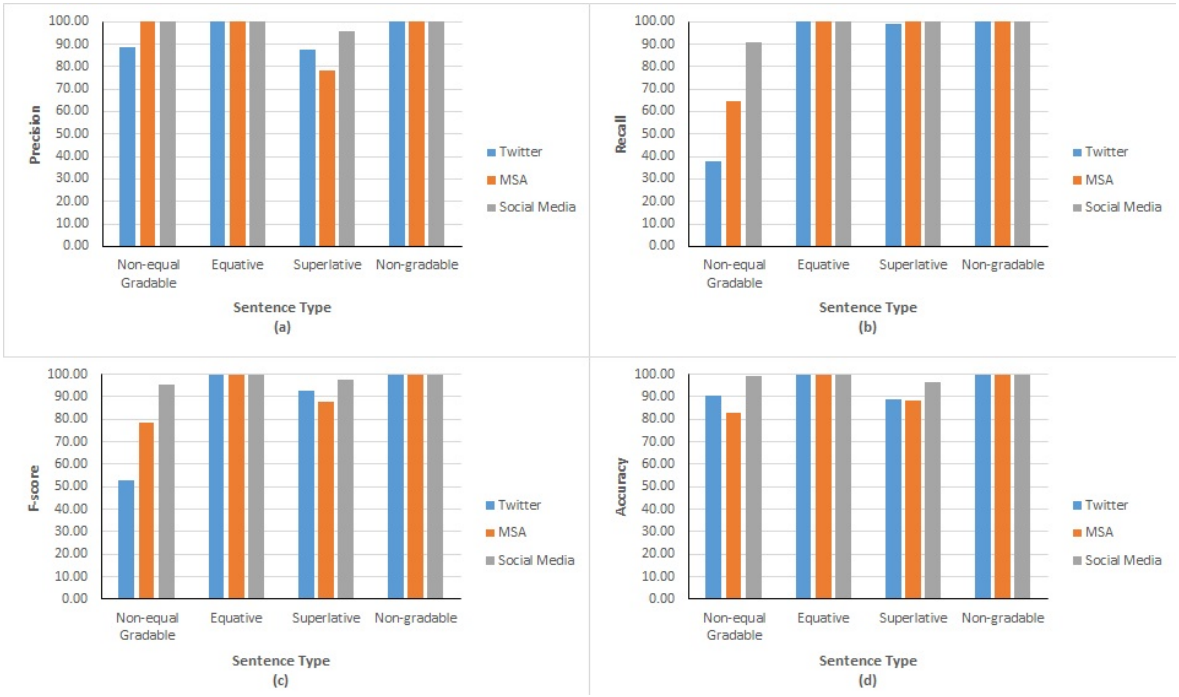

Figure 3. Graphical evaluation results of ACSTI algorithm (**a**) Precision, (**b**) Recall, (**c**) F-score, and (**d**) Accuracy.

Table 6. Numerical evaluation results of ACSTI algorithm.

Dataset	Sentence Type	Precision	Recall	F-Score	Accuracy
Twitter	Non-equal Gradable	88.89	37.65	52.89	90.69
	Equative	100	100	100	100
	Superlative	87.57	99.13	92.99	88.94
	Non-gradable	100	100	100	100
MSA	Non-equal Gradable	100	64.86	78.69	82.67
	Equative	100	100	100	100
	Superlative	78.38	100	87.88	88.57
	Non-gradable	100	100	100	100
Social Media	Non-equal Gradable	100	90.91	95.24	99.05
	Equative	100	100	100	100
	Superlative	95.78	100	97.85	96.76
	Non-gradable	100	100	100	100

5.2.1. Discussion of Non-Gradable Sentence Type Results

As shown previously in Figure 3, the evaluation results of the ACSTI algorithm for the non-gradable comparative sentence type show 100% truly identification percentage in all datasets. We would like to note that, to the best of our knowledge, this is considered the first research work made in Arabic non-gradable comparative sentence type. However, there are some limitations in this work. The ACSTI algorithm does not take into account the different versions of the same Arabic comparison keyword. For example, the Arabic non-gradable comparison keyword (أما) (while) can be wrote as (إما ـ اما) which have the same meaning. Also, the Arabic diacritics is not considered. For example, the Arabic word (أُمًا) (mother) can be falsely identified as a non-gradable comparison keyword.

5.2.2. Discussion of Superlative Sentence Type Results

As shown previously in Figure 3, the precision of proposed ACSTI algorithm for superlative sentence type is low in some datasets. This is due to some limitations in the Arabic language which may lead to such false identification results. For example, the Arabic superlative comparison keywords (جِبـ نِعْمَ) (the best) cannot be detected as comparative keywords if Arabic diacritics are not considered when analyzing the following superlative comparative sentence أحمد نِعْمَ الطالب الذكى (Ahmed is the best intelligent student). Also, in some Arabic sentences, the non-equal comparative sentence type is falsely identified as a superlative comparative sentence type. For example, the sentence موبايل أ افضل فى البطارية والكاميرا و أشياء أخرى كتيرة من موبايل ب (mobile A is better in battery, camera and other things than mobile B) can be falsely identified as a superlative comparative sentence while it is actually categorized as a non-equal one. This is because the relation keyword افضل (better) is located in the sentence faraway from the word من (than).

5.2.3. Discussion of Equative Sentence Type Results

As shown previously in Figure 3, the precision of the proposed ACSTI algorithm for equative sentence type are 100% using Twitter, MSA, and social media datasets. However, there are some limitations to the Arabic language which may cause false identification of equative sentence type. The two Arabic equative comparison keywords نفس and نفسه may provide other meanings not related to comparisons. For example, the word (نفس) can mean a person or a spirit in English. Also, the word نفسه can mean himself, a hope, or a wish in English. Therefore, the context of the sentence should be taken into consideration in order to consider such words as comparison keywords.

5.2.4. Discussion of Non-Equal Gradable Sentence Type Results

As shown previously in Figure 3, the precision of the ACSTI algorithm for the non-equal gradable sentence type were 88.9% using Twitter dataset and 100% using both MSA and social media datasets. However, there are some limitations in the Arabic language may cause false identification results for the Arabic non-equal comparative sentence type. Some of these limitations are discussed as follows:

- Some Arabic comparison keywords are used in the Arabic language as normal words not as comparison keywords. This is due to the misspelling of the comparison keywords. For example, in the following sentence اللهم أرحم من رحلو عنآ دون ودآع (may God have mercy on those who left us without saying goodbye), the comparison keyword here was supposed to be written (إرحم) (have mercy) but it was written (أرحم) (mercier). In this example, the sentence will be identified as a non-equal comparison sentence while it is not.
- Some Arabic comparison keywords can provide a meaning in the Arabic language that is completely different from the comparison meaning such as the Arabic keyword (أقل) (tell) in the sentence قل لى من أصحابك، أقل لك من أنت (if you told me who your friends are, I will tell you who you are). To resolve this issue the context of the sentence should be considered in the identification process.
- In some Arabic superlative sentences, like أحمد أفضل من عرفت فى حياتى (Ahmed is the best person I have known in my life), using the Arabic word من may force the sentence to be falsely identified as non-equal sentence type. In such a sentence, the word من means in English, a person not the word, than. Diacritics should be considered to obtain true identification results.

5.3. Evaluation of REACS, PEEANCS and PEEASCS Algorithms

This section presents the evaluation results of the proposed REACS, PEEANCS, and PEEASCS algorithms for the different comparative sentence types. Figure 4 and Table 7 show the precision of the three algorithms using each of the three considered datasets, respectively. As previously described, the proposed REACS algorithm extracts the following relation components from a comparative sentence if all or some of them are applicable to the sentence type. The components are the relation word, the extracted feature, entity 1, entity 2 and the preferred entity. Figure 4 and Table 7 show the precision of each of the relation components for the four types of the Arabic comparative sentences. The precision was calculated using Equation (1). The TP and FP parameters mentioned in the equation indicate the number of comparative sentences in which the relation components were truly and falsely identified, respectively. Each comparative sentence type has a separate subfigure in Figure 4.

The evaluation results of the REACS algorithm show that the relation word was truly identified by almost 100% in all Arabic comparative sentence types using all datasets. In general, the three evaluated algorithms provided the highest precision in the MSA dataset when compared with the other two datasets for all sentence types. This finding applies to almost all relation components. The detailed discussion of the low precision results that occur in Figure 4 are presented later in the next two sections.

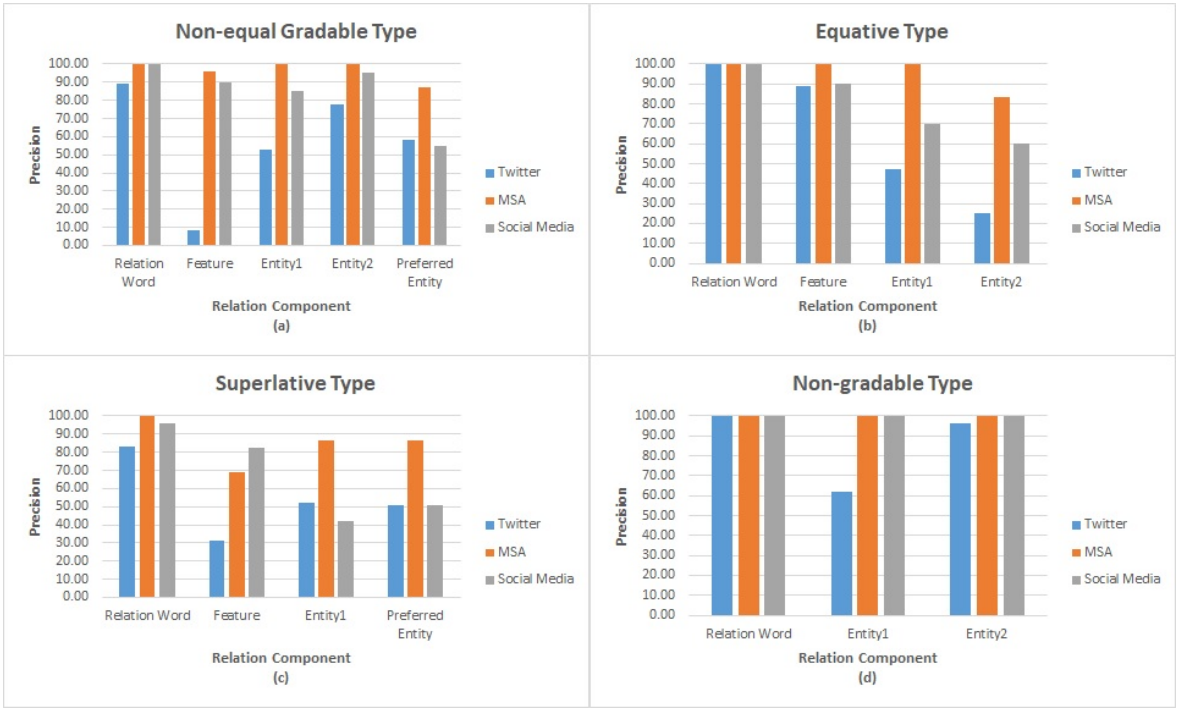

Figure 4. Graphical evaluation results of REACS, PEEANCS and PEEASCS algorithms for each comparative sentence type, (**a**) Non-equal gradable, (**b**) Equative, (**c**) Superlative, and (**d**) Non-gradable.

Table 7. Precisionof REACS , PEEANCS, and PEEASCS algorithms for each comparative sentence type.

Dataset	Sentence Type	Relation Word	Feature	Entity1	Entity2	Preferred Entity
Twitter	Non-equal Gradable	88.89	8.33	52.78	77.78	58.33
	Equative	100	88.89	47.22	25	N/A
	Superlative	82.79	31.36	52.01	N/A	50.67
	Non-gradable	100	N/A	62.07	96.55	N/A
MSA	Non-equal Gradable	100	95.83	100	100	87.5
	Equative	100	100	100	83.33	N/A
	Superlative	100	68.97	86.21	N/A	86.21
	Non-gradable	100	N/A	100	100	N/A
Social Media	Non-equal Gradable	100	90	85	95	55
	Equative	100	90	70	60	N/A
	Superlative	95.6	82.5	41.9	N/A	50.6
	Non-gradable	100	N/A	100	100	N/A

5.3.1. Discussion of REACS Algorithm Evaluation Results

There are some limitations to the Arabic language which may cause false identification results of the REACS algorithm. Such limitations include the following.

- In some Arabic language sentences, each of entity 1 and entity 2 consists of more than one word. Each entity can be formed of several words; however, it is considered only one element in the relation extraction process. This results in extracting false relation elements, i.e., entity 1, entity 2 and feature.
- The difference in the sentence order can affect the correct extraction of relation components. For example, in the superlative comparative sentence type, the relation word can exist in the beginning of the comparative sentence. An example of this in the sentence أول رخصة (the first license), the relation word is أول (the first) and entity 1 is رخصة (license). However, in other sentence, the relation word may exist in the middle of the sentence and entity 1 can exist in the beginning. For example, in the sentence البلاد فى أضعف حال (the countries are in the weakest situation), the relation word is أضعف (the weakest) and entity 1 is البلاد (the countries). A third example, is in the sentence أصعب ما فيها الطموح (the hardest thing about it is ambition), where the relation word is أصعب (the hardest) and entity 1 is الطموح (ambition), which exists by the end of the sentence.
- Entity 1 can be itself a comparative sentence and can exist by the end of the main comparative sentence such as the sentence أن الأقرب للواقع يكون الأدرى بالمصلحة (the closest to reality is the most knowledgeable of the benefits). In this sentence, the relation word of the main comparative sentence is الأقرب (the closest) and entity 1 is الأدرى بالمصلحة (the most knowledgeable of the benefits).
- The correct relation extraction from some Arabic sentences mainly depends on precise understanding of the meaning and the context of the sentence such as أغنية أجمل سنين عمرنا (the song named "the most wonderful years in our life"). In this sentence, the relation word is أجمل and entity 1 is سنين عمرنا ("the most wonderful years in our life") not أغنية (the song).

5.3.2. Discussion of PEEANCS and PEEASCS Evaluation Results

This section discusses some of the limitations of extracting the preferred entity from the Arabic comparative sentence. The correct extraction of the preferred entity has a big challenge, which depends on the meaning of the relation keyword and the extracted feature. For example, in the two sentences, المكان الأكثر جمالا (the most beautiful place) and المكان الأكثر قبحا (the most ugly place), the word المكان (place) is the entity 1. This word represents the preferred entity in the former sentence, while it is not the preferred entity in the later sentence. This is because the feature in the former sentence is a positive word, i.e., جمالا (most beautiful), while in the later sentence, the feature is a negative word, i.e., قبحا (ugly), although the relation keyword, i.e., الأكثر (most) in both sentences has a positive sentiment.

Another limitation of the correct extraction of the preferred entity from the Arabic comparative sentence is the presence of the Arabic negation words, Such words convert entity 1 from a preferred entity to a non-preferred one. This occurs although the relation keyword has a positive sentiment. For example, in the sentence ليست أفضل من (not better than), the negation word ليست (not) affects the correct extraction of entity 1 as the preferred entity.

6. Conclusions and Future Work

This paper proposed a lexicon-based framework for the detailed analysis of Arabic comparative sentences. The proposed framework comprises a set of proposed algorithms for identifying comparative sentences, identifying comparative sentences types, extracting relation components and extracting preferred entity from Arabic comparative sentences. The proposed algorithms were abbreviated as ACSI, ACSTI, REACS, PEEANCS and PEEASCS. The proposed framework was evaluated using three Arabic datasets; one of them is publicly available, while two of them were manually developed for this purpose. The proposed framework leverages a lexicon of Arabic comparative keywords that was specifically developed to operate and evaluate the framework. This lexicon contains 649 Arabic comparative keywords that covers all the Arabic comparative sentence types.

The evaluation metrics used in the experiments were the four standard metrics, namely precision, recall, F-score, and accuracy. The average accuracy value of the proposed ASCI algorithm was 97% across all datasets. The ACSI algorithm evaluation results showed that the average values of the four evaluation metrics over all datasets were in the range of 92% to 97%. The ACSTI algorithm evaluation results showed that the average values of the evaluation metrics of the proposed type identification algorithm over all comparative sentence types using all datasets were in the range of 91% to 96%. The evaluation results of the REACS, PEEANCS and PEEASCS algorithms showed that the average results over all datasets were 97% precision for relation word extraction, 73% precision for feature extraction, 75% precision for first entity extraction, 82% precision for second entity extraction, and 65% precision for preferred entity extraction.

Future work includes a set of extensions. Firstly, Arabic comparative sentences with more than one Arabic comparative keyword can be investigated. This extension requires incorporating an additional preprocessing step of a divide-and-conquer-like process. In this process, complex sentences with more than one Arabic comparative keyword should be detected and divided into a set of independent simple sentences. Each of these sentences can then be separately investigated by using the proposed framework. Afterwards, the results obtained from the investigation of each simple sentence can be combined to provide a summarized evaluation result of the complex sentence.

Secondly, additional singular, plural and feminine Arabic comparison keywords can be added to the comparative keywords lexicon in addition to taking into consideration Arabic diacritics when manually classifying comparison keywords. This can be supported by the employment of Arabic language experts to formally guide this manual classification. Thirdly, Arabic features dictionary of positive and negative features can be used to detect the Arabic negation words so that the preferred entity can be correctly extracted. Fourthly, additional Arabic comparison keywords can be used to handle their corresponding comparative sentence types in the proposed approach. Fifthly, lexicons for other popular Arabic dialects like Moroccan or Iraqi can be developed to increase the universality of the proposed framework among other Arabic dialects. Finally, the proposed framework can be compared with more advanced machine and deep learning approaches [9,10,13,22,23]. Transformer-based natural language processing models like AraBERT [24,25] can be used to take into consideration the context of the sentence, not only the comparison keyword.

Author Contributions: Conceptualization, A.H. and A.Y.; methodology, A.H. and A.Y.; software, A.H.; validation, A.H. and A.Y.; formal analysis, A.H. and A.Y.; investigation, A.H.; data curation, A.H.; writing—original draft preparation, A.H.; writing—review and editing, A.H. and A.Y.; visualization, A.H. and A.Y.; supervision, A.K. and A.Y.; project administration, A.K. and A.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACSI	Arabic Comparative Sentence Identification
ACSTI	Arabic Comparative Sentence Type Identification
REACS	Relation Extraction from Arabic Comparative Sentence
PEEANCS	Preferred Entity Extraction from Arabic Non-Equal Comparative Sentence
PEEASCS	Preferred Entity Extraction from Arabic Superlative Comparative Sentence

References

1. El-Halees, A.M. Opinion mining from Arabic comparative sentences. In Proceedings of the 13th International Arab Conference on Information Technology ACIT, Balamand, Lebanon, 11–13 December 2012.
2. Sakr, A.M.; Keshk, A.; Youssef, A. Analysis and Mining of Arabic Comparative Sentences: A Literature Review. *IJCI Int. J. Comput. Inf.* **2024**, *11*, 66–78. [CrossRef]
3. Alharbi, F.R.; Khan, M.B. Identifying comparative opinions in Arabic text in social media using machine learning techniques. *SN Appl. Sci.* **2019**, *1*, 1–13. [CrossRef]
4. El Defrawi, M.; Salah, M.; Abd Al-Aziz, A.; Eldin, A.S. Comparative relation extraction from Arabic opinions. *Int. J. Comput. Sci. Inf Secur.* **2017**, *15*, 230–235.
5. Bach, N.X.; Van Pham, D.; Tai, N.D.; Phuong, T.M. Mining Vietnamese comparative sentences for sentiment analysis. In Proceedings of the 2015 Seventh International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, Vietnam, 8–10 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 162–167.
6. Eldefrawi, M.M.; Elzanfaly, D.S.; Farhan, M.S.; Eldin, A.S. Sentiment analysis of Arabic comparative opinions. *SN Appl. Sci.* **2019**, *1*, 1–11. [CrossRef]
7. Al-Sawi, L.; Saad, I. *Al-Murshid: A Guide to Modern Standard Arabic Grammar for the Intermediate Level*; Amer Univ in Cairo Press: Cairo, Egypt, 2012.

8. DoniaGamal, M.A.; El-Horbaty, E.S.M.; Salem, A.B. Opinion mining for Arabic dialects on twitter. *Egypt. Comput. Sci. J.* **2018**, *42*.
9. Younis, U.; Asghar, M.Z.; Khan, A.; Khan, A.; Iqbal, J.; Jillani, N. Applying machine learning techniques for performing comparative opinion mining. *Open Comput. Sci.* **2020**, *10*, 461–477. [CrossRef]
10. Khan, A.; Younis, U.; Kundi, A.S.; Asghar, M.Z.; Ullah, I.; Aslam, N.; Ahmed, I. Sentiment classification of user reviews using supervised learning techniques with comparative opinion mining perspective. In Proceedings of the Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Las Vegas, NV, USA, 2–3 May 2019; Springer: Berlin/Heidelberg, Germany, 2020; Volume 21, pp. 23–29.
11. Yang, S.; Ko, Y. Classifying Korean comparative sentences for comparison analysis. *Nat. Lang. Eng.* **2014**, *20*, 557–581. [CrossRef]
12. Liu, Q.; Huang, H.; Zhang, C.; Chen, Z.; Chen, J. Chinese comparative sentence identification based on the combination of rules and statistics. In Proceedings of the Advanced Data Mining and Applications: 9th International Conference, ADMA 2013, Hangzhou, China, 14–16 December 2013; Proceedings, Part II 9; Springer: Berlin/Heidelberg, Germany, 2013; pp. 300–310.
13. Alotaibi, N.; Al-onazi, B.B.; Nour, M.K.; Mohamed, A.; Motwakel, A.; Mohammed, G.P.; Yaseen, I.; Rizwanullah, M. Political Optimizer with Probabilistic Neural Network-Based Arabic Comparative Opinion Mining. *Intell. Autom. Soft Comput.* **2023**, *36*, 3121–3137. [CrossRef]
14. Nabil, M.; Aly, M.; Atiya, A. Astd: Arabic sentiment tweets dataset. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2515–2519.
15. Martinez, A.R. Part-of-speech tagging. *Wiley Interdiscip. Rev. Comput. Stat.* **2012**, *4*, 107–113. [CrossRef]
16. Berrar, D. Bayes' theorem and naive Bayes classifier. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 403.
17. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", Sicily, Italy, 3–7 November 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 986–996.
18. Bonaccorso, G. *Machine Learning Algorithms*; Packt Publishing Ltd: Birmingham, UK, 2017.
19. Sutton, C.; McCallum, A. An introduction to conditional random fields. *Found. Trends® Mach. Learn.* **2012**, *4*, 267–373. [CrossRef]
20. Wallach, H.M. Conditional Random Fields: An Introduction. Technical Reports (CIS). 2004; p. 22. Available online: https://www.inference.org.uk/hmw26/papers/crf_intro.pdf (accessed on 5 January 2025).
21. Dalianis, H. Evaluation Metrics and Evaluation. In *Clinical Text Mining: Secondary Use of Electronic Patient Records*; Springer International Publishing: Cham, Switzerland, 2018; pp. 45–53.
22. Bayazed, A.; Almagrabi, H.; Alahmadi, D.; Alghamdi, H. ACOM: Arabic Comparative Opinion Mining in Social Media Utilizing Word Embedding, Deep Learning Model & LLM-GPT. *IEEE Access* **2024**, *12*, 148741–148755.
23. Setyanto, A.; Laksito, A.; Alarfaj, F.; Alreshoodi, M.; Oyong, I.; Hayaty, M.; Alomair, A.; Almusallam, N.; Kurniasari, L. Arabic language opinion mining based on long short-term memory (LSTM). *Appl. Sci.* **2022**, *12*, 4140. [CrossRef]
24. Antoun, W.; Baly, F.; Hajj, H. Arabert: Transformer-based model for arabic language understanding. *arXiv* **2020**, arXiv:2003.00104.
25. Abdul-Mageed, M.; Elmadany, A.; Nagoudi, E.M.B. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. *arXiv* **2020**, arXiv:2101.01785.