International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 1, February 2023, pp. 746~755
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i1.pp746-755

AgroSupportAnalytics: big data recommender system for agricultural farmer complaints in Egypt

Esraa Rslan[1], Mohamed H. Khafagy[1], Mostafa Ali[1], Kamran Munir[2], Rasha M. Badry[1]
[1]Faculty of Computers and Information, Fayoum University, Faiyum, Egypt
[2]Department of Computer Science and Creative Technologies, University of West of England, Bristol, United Kingdom

Article Info

Article history:

Received Nov 15, 2021
Revised Jul 15, 2022
Accepted Aug 18, 2022

Keywords:

Agricultural recommender system
Latent semantic analysis
Semantic textual similarity
Support vector machine classification

ABSTRACT

The world's agricultural needs are growing with the pace of increase in its population. Agricultural farmers play a vital role in our society by helping us in fulfilling our basic food needs. So, we need to support farmers to keep up their great work, even in difficult times such as the coronavirus disease (COVID-19) outbreak, which causes hard regulations like lockdowns, curfews, and social distancing procedures. In this article, we propose the development of a recommender system that assists in giving advice, support, and solutions for the farmers' agricultural related complaints (or queries). The proposed system is based on the latent semantic analysis (LSA) approach to find the key semantic features of words used in agricultural complaints and their solutions. Further, it proposes to use the support vector machine (SVM) algorithm with Hadoop to classify the large agriculture dataset over Map/Reduce framework. The results show that a semantic-based classification system and filtering methods can improve the recommender system. Our proposed system outperformed the existing interest recommendation models with an accuracy of 87%.

Corresponding Author:

Esraa Rslan
Faculty of Computers and Information, Fayoum University
Faiyum, Egypt
Email: esraa.rslan@fayoum.edu.eg

1. INTRODUCTION

It is expected in the future, big data applications will be widely used by farmers, experts, and others across the agricultural industry [1]. However, despite the huge amount of data already generated over thousands of agricultural farms each year in Egypt, the impact of big data is still incomplete. While the variety, velocity, volume, and data generated in the agriculture process have been available, the advantages of aggregation, analysis, and distilling value-creating decision support tools from that data remain in the early phases [2]. Agriculture plays an essential role in the country's economy. However, When the coronavirus paramedic happens, and because of the social distance, it is hard for farmers to interact or contact agricultural specialists to get suitable solutions to the different agriculture problems according to crop type [3]. One way of treating this virus is eating healthy food, which is essential for energy and crucial to defeating the disease. The shortage of different support for farmers to achieve good agricultural practices and prevent methods is another metric that hinders food productivity. Farmers need quick advice on plant diseases, seed patterns, and prevention methods to face environmental changes. However, farmers' access to such information is highly minimized to the support systems being incompatible, unreliable, and predominately not certain, so delivered advice becomes incorrect [1], [4].

The farmers submit their problems, then the AgroSupportAnalytics [5] recommend and suggest a suitable solution for the farmer's complaint. The AgroSupportAnalytics aimed to solute the problem of support and provide recommendations for farmers in Egypt. Agricultural problems are divided over 4,242 villages in Egypt [6]. In Arabic text, the problems are collected and submitted to one of the 198 centers spread over the state to support farmers in their regions. Storage of agriculture complaints and solutions has been made on a public cloud that hosts analytics observations and support toolkits [7].

The proposed AgroSupportAnalytic system developed a support vector machine (SVM) classification method for the used agricultural dataset with Hadoop Map (M)/Reduce (R) in the parallel environment. In the AgroSupportAnalytic system, we classify the agriculture dataset and latent semantic analysis (LSA) semantic similarity for semantic analysis [8], [9]. Map/Reduce approach provides a fast implementation of classification steps in large datasets and is a powerful big data analysis tool. Dataset is saved on the cloud, and it contains 10,000 complaint problems. In addition, the proposed system used the LSA [10] to measure the semantic similarity among the farmer problems and the available agriculture problems in the used dataset.

Due to increasing the text data available in different languages, many research papers focuses on semantic similarity measures across languages. In the work of semantic similarity in Arabic-English texts, the authors [11], [12] used Latent Semantic Indexing in semantic Arabic-English language to compute the semantic similarity between Arabic text and the English one. Alzahrani [13] introduced two Semantic Similarity methods for Cross-Language Arabic English Sentences (CLAES). The author used a dictionary translator in the first method, as an Arabic sentence is translated into English. After that, the semantic similarity is calculated by applying translation similarity techniques. The second method, Machine translation, is used for the Arabic sentence. Potthast *et al.* [14] discussed the Cross-Language Plagiarism Detection of Arabic-English documents. First, the system translates the text by retrieving all the available translations of synonyms for a word from WordNet [15], then applying keyphrase extraction. Finally, a combination of monolingual is calculated (Cosine similarity, N-Gram, and longest common subsequence (LCS)) to return similar sentences. These methods achieve great results with languages that are near in meaning to each other because of joint root words. However, measuring the semantic similarity could be more complicated if the languages are different. Dai and Huang [16], for example, computed the semantic word similarity for applications in the cross-language semantic space. They measured the similarities between two texts, one in the Chinese language and the other in English. Zou *et al.* [17] introduced a technique that extracts the main features of mono and cross-lingual semantic relations across different languages. They proposed a method storing the bilingual embedding between Chinese and English from a large corpus. Also, machine translation is used to align between words.

Processing large text is a challenging task, especially in text analysis. Map/Reduce is based on a distributed and parallel framework for utilizing several tasks. Such as text processing tasks, dividing data and computation loads in a cluster, text clustering, information extraction, storing, fetching unstructured data [18], natural language processing, text summarization, and sentence similarity [19], [20]. Text similarity is an extensively challenging problem in text analysis. Many techniques are proposed for handling large text for automatic text summarization. Nagwani [21] introduced a Map/Reduce framework of multi-document for text summarization. Many types of research are concerned with implementing techniques, algorithms, and approaches in parallel environments like Hadoop and Cloudera [22]–[24]. Hadoop is an Apache-based framework used to analyze massive data sets on clusters containing many machines, using Map and Reduce approach. Hadoop Map/Reduce allows applications to run in parallel environments. Many papers on the support vector machine algorithm in parallel machines are proposed.

In the proposed system, first, the farmer writes the agriculture complaint in the Arabic script; then, Google's machine translation is used to translate the complaint from Arabic into English. Second, analyzing the complaints using data analytics techniques to retrieve term frequency and classify the complaint to which problem class using SVM in Map/Reduce technique. Third, returning a recommended answer by searching for similar complaints in the agriculture historical dataset. The recommended response uses LSA [25] to measure the semantic similarity process between cross-language in Arabic-English sentences. There are two methods used in the LSA algorithm. The first method is applied using term frequency weighting (TF), while the second is based on inverse document frequency (TF-IDF). LSA can get much better results than the different plain vector space models. It creates a decomposing term-document matrix, since it is faster than other dimensional reduction methods. However, when the data representation is dense, it is hard to index words based on particular keywords. It works well on a dataset with diverse topics. Moreover, LSA can handle synonymy problems based on the dataset. Applying LSA on new data is easier and faster compared to other methods as the matrix in topics space has to be multiplied with the TF-IDF vector to get the latent vector of a document.

The paper is as follows: in section 2, research method, we explain our proposed system LSA with SVM classification in Map/Reduce. The results and discussion section are explained in section 3. Finally, the conclusion section is presented in section 4.

2. RESEARCH METHOD

This section presents the AgroSupportAnalytics system. It has five steps. The main steps are machine translation, preprocessing, SVM Map/Reduce classifier, feature extraction, and finally, applying LSA. The system is presented in Figure 1.

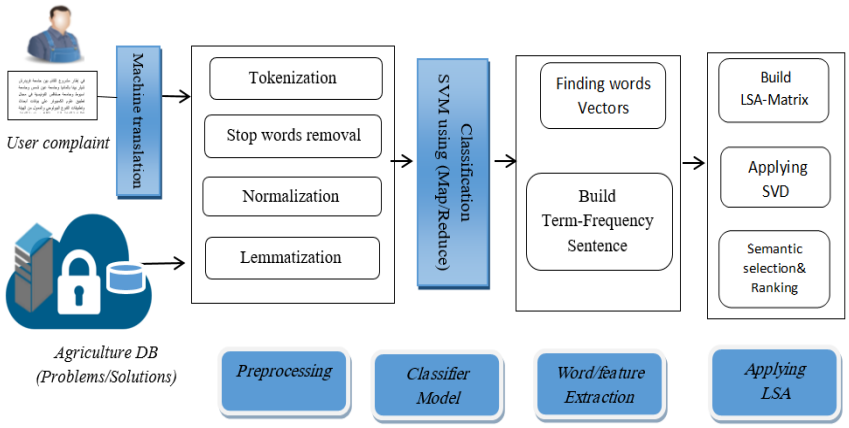

Figure 1. The proposed system phases

2.1. Machine translation

The farmers write their complaints in Arabic script. Therefore, these complaints need to be translated into English as the agriculture dataset is located over the cloud in the English language. Google's Cloud Translator API is used to translate the Arabic complaints into English. Google Cloud Translate has high accuracy also considered more reliable [26].

2.2. Preprocessing

The farmer may write their problems with more details containing the plant age, planting method, watering method, disease description, and soil type. Such problems may be written in undesired form and un-understandable meaning or may contain useful words that affect the text processing of model phases. Preprocessing is important for bringing the farmer query and the historical complaint/solution data into a form that similar task can be processed with the farmer query [27]. Data preprocessing includes processes: tokenization, stop words removal, normalization, and lemmatization.

2.2.1. Tokenization

Tokenization is breaking up sentences into pieces such as keywords that pieces called tokens. Words are separated by blanks like white space, semicolons, commas, and quotations. In tokenization, some characters like punctuation marks, unique characters, and white spaces are removed [28]. This method minimizes text data processing and improves system performance.

2.2.2. Stop words removal

The process of stop word removal is removing all words that don't have any meaning. Stop words must be eliminated from farmer queries since they don't have any effect or importance in the sentences' meaning. Examples of stop words in the English language are "am", "is", "are", "the", and "a". We used the WordNet database to get the list of all stop words.

2.2.3. Normalization

This task is important for noisy texts; it focuses on removing unwanted data or characters in the query and historical dataset, like repeating words, text lowercase, and spaces. An example of some characters is "÷,×,؛,<,>,_,(,),,,],؟". So, for example, the word "croop" can be transformed into" crop " standard form.

2.2.4. Lemmatization

The process of lemmatization is finding the root (base form or lemma) of a word by considering its inflected forms like “appearance, appearing” have the same root “appear”. For example, the lemmatization brings "mice" and “mouse” both as “mouse”. We used an NLTK lemmatizer with POS tags.

2.3. Classification

This phase aims to group the similar problems to make them ready for text similarity and ensure that all similar problems in the dataset participate as a group in the similarity process. The dataset is classified into problem categories like the pest, weed, diseases, and irrigation. Using Map/Reduce framework in the proposed system, text processing speed and scalability are improved compared to other traditional systems. In the classification process, the SVM Map/Reduce approach is applied to the agriculture dataset and the farmer query. In the given set of training problems from the dataset, each one belongs to one of the main categories. SVM algorithm builds a model that specifies new farmer problems into one category or another, working as a non-probabilistic binary linear classifier. SVM model represents the agriculture problems as it points in space, so the examples of the separate categories are split by a clear gap that is broad as possible. Then, new queries are mapped into the same space and predicted the category based on which side of the gap they fall on. This classification process is performed in a parallel manner by parallelizing the classification process in several machines. SVM is one of the most important classification algorithms that work effectively on many high-dimensional tasks. Accuracy is mostly high; reliable results when training classes have errors; speed evaluation of the learned function. However, SVM algorithm might have a long training time; because it is not easy to learn the function weights.

Hadoop Map/Reduce is applied to classify the farmer query based on which problem class belongs to find the suitable solution. The innovation of Hadoop is that there is no need for expensive tools. In the state, it distributes large amounts of data on several machines with high reliability and scalability for data storage and processing. Map/Reduce is the main concept of big data. It is a programming method that allows extensive agriculture problems to be divided between multiple machines in a Hadoop. After this step, each complaint is classified according to the problem category (weeds, irrigation, pest, diseases category). The classification phase will help to increase the performance of the semantic similarity process and the system efficiency.

2.3.1. SVM in parallel network environment

Transferring each complaint in the dataset into one vector in the parallel environment having 2 phases: Map (M) and Reduce (R) phases. The input of the Map phase is one complaint, and the output is many components of a vector corresponding to the sentences of the text. In the Map phase, we transfer the text into one vector, similar to SVM input of the Reduce is the output of the Map phase, and it has many portions of a vector. The output of the Reduce is a vector that corresponds to the sentences in the text. In the Reduce phase, those vector components are merged into one vector.

2.3.2. Hadoop map (M)

The n vectors of one complaint are input into the Hadoop Map (M). Then, the SVM algorithm is performed to cluster, where every vector of n vectors of one text complaint in the testing dataset. The output is the result of classifying the vector into weed vector set, pest vector set, diseases vector set, or irrigation vector set.

2.3.3. Hadoop reduce (R)

The classification results of the n vectors into the problem category vector group in the Hadoop Map (M) phase are input into the Hadoop Reduce (R) in the parallel network environment. Then, in the Reduce (R) phase, the testing dataset's polarity of one complaint (corresponding to the n vectors) is specified correctly.

2.4. Word Extraction

The feature extraction process is generally utilized in text-similarity applications. The extraction process calculates the appearance of important word features in a text to construct the word vector. We use the extracted features from a group of sentences to give a value for each problem in the dataset. This process helps to construct a term frequency matrix. In this paper, algorithms are explained with their time complexity. Algorithm 1 is utilized to construct the sentence vector from word vectors with TF-IDF weights. The main procedure in this algorithm is in the first inner for loop, lines *three–to eleven*, the sentence vector is built for every sentence in the farmer problem. Let N be the number of words in the problem, M be the number of words in the sentence, and $|P|$ is the number of sentences in whole problem. The algorithm calculates the word vector for each word in all sentences in the problem, so there is an execution time

complexity of *O* (M.|*P*|). When using the TF-IDF to weight the word vectors, the score of TF-IDF for every word will be calculated in the same loop. The algorithm requires to visit each word in the sentence only one time where N= M.|*P*| and the time complexity is *O* (N).

Algorithm 1. Build sentence vector

```
Input: sentences problem P.
Output: vectors of the sentences Pv.
Begin
    1.  Step 1: For every sentence si in P do
    2.  Step 2: mi:=0
    3.  Step:3 For every word w in sentence si do
    4.    wv:=word2vec(w)
    5.        If exist (wv) then
    6.        tf_id_fscorew:=TF-IDF(w)
    7.        wv:=wv*tf_id_fscorew
    8.        mi +:=1
    9.        svi +:=wv
    10.      End if
    11. End For
    12. svi: = svi /mi
    13. End For
End
```

2.5. Applying LSA

Word vectors are created after preprocessing and classifying the farmer text and agriculture dataset. Then, we apply LSA [29], [30] to calculate the semantic similarity between the farmer text and the available agriculture dataset. LSA is a powerful corpus-based technique for calculating semantic similarity. It consists of three steps are input matrix creation, singular value decomposition (SVD), and sentence selection.

2.5.1. Term-sentence matrix

An input matrix is built for the farmer's complaint and historical dataset. Every row in the matrix represents the word or term in the farmer's complaint and agriculture dataset. Every column represents the complaint. The cell value is the intersection between term and complaint. Two methods of weighting schema are utilized to fill the cell values: TF-IDF or TF. Thence, we choose the sentences with important attributes using the most frequent term. TF-IDF is one of the methods to rank the most frequent terms. It is a statistical method used to know how a term occurs in a sentence. The first part is TF, constant for all term weighting methods, and calculated as shown in (1):

$$\mathrm{TF_{ij}} = \log(tf_{\mathrm{ij}} + 1); \quad tf_{ij} = \frac{n_{ij}}{N_j} \tag{1}$$

where n_{ij} is the number of times the i^{th} word exists in j^{th} complaint, N_j is the complaint size (number of words in the complaint). The second section of the term weighting is calculated once.

In TF-IDF, The Inverse Document Frequency (IDF) represents how many times a term T occurs in all problems of a text. The cells are filled with the weight of (TF-IDF) of a term (*i*) in the complaint according to (2):

$$\mathrm{TF-IDF_{ij}} = \mathrm{TF_{ij}} * \log\frac{|D|}{n_i} \tag{2}'$$

where TF_{ij} is the frequency of a term (*i*) in each complaint/problem (*j*), and IDF_{ij} =$\log\frac{|D|}{N_i}$ where |D| is the number of complaints in the dataset and ni is the number of complaints with the ith word.

2.5.2. Singular value decomposition

Singular value decomposition (SVD) is an algebraic matrix that plays an important part in identifying the relationships between words and sentences. It enhances the term sentence matrix and identifies the relations between terms and complaints [20]. SVD decomposes the term sentence matrix into three matrices that determine all the significant attributes of the matrices. After input matrix creation, SVD matrix X is constructed, which is the multiplication of three matrices, where the columns and rows are two vectors matrices built from eigenvalues, and the third one is a diagonal matrix. The matrix is calculated based on TF-IDF in the word frequency. Since the TF-IDF method has the primary metric to extract the most descriptive terms in a sentence and can compute the similarity between two sentences. The SVD can be presented using (3):

$$SVD = S\Sigma U^T \quad (3)$$

where S is the eigenvector of the multiplication of the matrix and the transpose $X^T(XX^T)$, Σ is the square root of the eigenvalue of (X^TX), and U^T is the eigenvector of the multiplication of the transpose X^T by the matrix X (X^TX). SVD minimizes the number of columns while remaining the number of rows, keeping the similarity matrix between the words. Every word has a value corresponding to its rows represented as a vector, and the cosine semantic similarity is measured between these vectors' values in the next phases.

2.5.3. Semantic selection and ranking

After applying the SVD, the cosine similarity is calculated between user complaint and each agriculture problem to return the correct solution. The cosine [31] can be calculated as (4):

$$\text{cosine similarity}(V1, V2) = \frac{V1.V2}{\|V1\|.\|V2\|} \quad (4)$$

where $cosine\ similarity(V1, V2)$ is the similarity between the farmer query and agriculture complaints dataset, *V1* is the weight of the term in the farmer query, and *V2* is the vector weight of the term in the complaints dataset. Finally, the complaints are ranked corresponding to the semantic similarity score. If the score is more than a specific threshold (75%), the system returns the response (answer) with the highest score corresponding to the best matching complaint. Finally, the system retrieves the recommended solution for the query farmer complaint.

Algorithm 2 constructs the similarity matrix between each two-sentence vector built from farmer problem and agriculture historical dataset based on the problem category. Its complexity relies on the execution time of the internal loop (*lines two-five*). This loop mainly calculates the similarity between each sentence's vectors with other vectors. The overall time complexity is estimated as $O\ (|V|^2)$.

Algorithm 2. Building similarity matrix

```
Input: Sentence Vectors set SV:=(sv1, sv2, sv3, …, svn)
Output: Similarity Matrix
Begin
    1.  Step 1: For m:=0 to |SV| do
    2.   Step 2: For k:=0 to |SV| do
    3.     if m!=k then
    4.  Step 3: SimilarityMatrix:=CosineSimilarity (sv1m, sv1k)
    5.     End if
    6.   End For
    7.  End For
End
```

3. RESULTS AND DISCUSSION

This section introduces an evaluation of the proposed recommender system using TF-IDF and TF. We measured the system performance in terms of precision, recall, F-measure, and accuracy. Our AgroSupportAnalytics system was implemented with python language. The dataset was divided into 80% training and 20% testing with ten experiments. Finally, we test the system with the test dataset and save the results of SVM classification. The experiments are executed in dual-core processor systems with a Pentium CPU speed of 6.00 GHz, GPU Tesla 16 GB, and 32 GB RAM. The systems (up to 4 nodes) are connected over a 100 Mbps LAN and the Windows XP (using MS-DOS Prompt).

3.1. Dataset

The dataset acquired from Egypt's agriculture research center (ARC) and virtual extension and research communication network (VERCON) [6] contains historical complaints and solutions provided by the experts saved as unstructured data. The agricultural data was installed on a public Cloud. This dataset is important because it has real-world problems collected over a long time by Egypt's agriculture centers. The dataset has different crop types like wheat, tomato, cotton, and mango, also problem categories like irrigation, pest, weed, and diseases. Table 1 shows statistics about the VERCON agriculture dataset. It lists the crops which are planted in Egypt. Also, the dataset is available in text form.

3.2. Experiments and results

The conducted experiments and results are presented to evaluate the system's performance with different measures. We applied experiments with two settings without/with classification techniques. Recommendations are returned on recommender techniques with classification and semantic similarity; the

result of SVM classification on the agriculture dataset is corporate to the recommendation process. We tested two semantic similarity methods for semantic analysis: TF and TF-IDF.

Consider the farmer query example: "*ظهور بقع بنية على محصول الزيتون*". The proposed system is applied to return the most relevant complaint and its solution for the farmer query, as shown in Table 2. First, the system translates the Arabic farmer query into English: “Appearance of brown spots on the olive crop”. Second, we apply preprocessing on the farmer query like tokenization, stop word removal and lemmatization. Third, apply classification by Map/Reduce SVM algorithm using Hadoop to classify farmers' query based on problem category “weed class”. Fourth create a term frequency matrix. Fifth, compute the semantic similarity score from the LSA matrix using TF-IDF or TF to return the most recommended solution.

Some metrics are used to evaluate the classification in our system. The accuracy is measured to know the accuracy of the classification results before semantic analysis. SVM is also applied to predict the farmer query belongs to which category before using recommendation methods. The results show that classification performance with accuracy is approximately 88%~89%, as shown in Table 3.

Table 1. VERCON agriculture dataset description

Summary	Description
Number of problems	10,000
Number of crops	40
Main problem category	Weed, pest, diseases, irrigation
Number of words/problems	15~20
Language	English

Table 2. An example of system steps

Process	An example
Translation	Appearance of brown spots on the olive crop
Tokenization	Appearance, of, brow, spots, on, the, olive, crop
Stop word removal	Appearance, brown, spots, olive, crop
Lemmatization	appear, brown spot on the olive crop
Classification	weed class
Solution	Sprinkling with zinc sulfate at a rate of 2 kg per 200 liters of water for 10 days, making sure to wash and clean the motor before spraying.

Table 3. The results of the SVM classification in dataset

Category	# Of records in dataset	Correct classification	Incorrect classification	Accuracy
Weed	2312	2035	277	88.02%
Pest	3150	2813	337	89.30%
Diseases	3566	3140	426	88.05%
Irrigation	1013	901	112	88.94%
Summary	10,041	8889	1152	88.52%

We applied experiments with two different settings with/without applying the classification technique to evaluate how the SVM classifier-based model enhances the system performance. We used different measurers such as Precision, Recall, F1-score, and accuracy in calculating the results of our system. As shown in Table 4, we used TF semantic similarity in our system. As a result, the F1-score is 83.82% using SVM classification and 69.94% without SVM classification, and the accuracy is 84.30% using SVM classification and 70.32% without SVM classification. It is noticed in Table 5, TF-IDF semantic similarity is used in our system. As a result, the F1-score is 86.64% using SVM classification and 73.42% without SVM classification, and the accuracy is 86.98% using SVM classification and 70.32% without SVM classification.

Table 4. The semantic evaluation results of using TF

Measures	With SVM classification TF	Without SVM classification TF
Precision	83.23%	71.24%
Recall	84.41%	68.65%
F1-score	83.82%	69.94%
Accuracy	84.30%	70.32%

Table 5. The semantic evaluation results of using TF-IDF

Measures	With SVM classification TF-IDF	Without SVM classification TF-IDF
Precision	85.91%	74.63%
Recall	87.17%	72.21%
F1-score	86.64%	73.42%
Accuracy	86.98%	74.01%

Different measures are used to evaluate the performance of the proposed recommender system, such as the root-mean-square error (RMSE), mean absolute error (MAE), and normal MAE (NMAE). RMSE, MAE, and NMAE are well-known metrics used as a baseline to evaluate the recommender system. Table 6 shows the comparative results acquired from the recommender using these metrics with semantic analysis. They were calculated based on SVM classification with the four main problem categories. Recommendations are based on recommender system methods with classification and semantic similarity. Table 6 shows that RSME, MAE, and NMAE yielded by the system that merges SVM classification with semantic similarity are better than the error rates obtained by methods without SVM classification.

We concluded from both Table 4 and Table 5 that using LSA with different methods in the system. Figures 2 illustrates the comparison of the semantic similarity-based methods (TF, TF-IDF) With SVM classification in Figure 2(a) and Without SVM classification in Figure 2(b). The system achieved an accuracy average of 84.30% in TF, while TF-IDF scores a better accuracy of 86.98% With SVM classification.

Table 6. RMSE, MAE, and NMAE values with different categories

	With SVM classification				Without SVM classification			
	Weed	Pest	Diseases	Irrigation	Weed	Pest	Diseases	Irrigation
RMSE	0.962	0.953	0.897	0.923	0.987	0.979	0.936	0.968
MAE	0.798	0.738	0.693	0.712	0.821	0.796	0.793	0.763
NMAE	0.2427	0.2113	0.235	0.2197	0.326	0.324	0.312	0.291

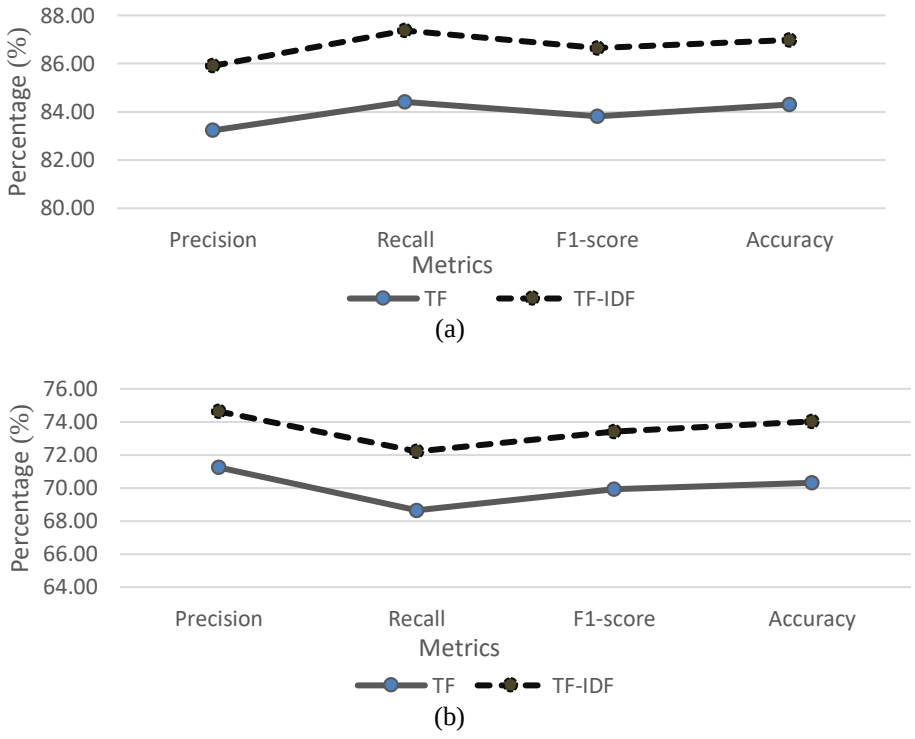

Figure 2. Comparing system results when using TF or TF-IDF: (a) with classification and (b) without classification

4. CONCLUSION

The main idea of this work is to find a solution to the farmers' problem by identifying the main causes of complaints and the features behind this. We developed a recommender system using LSA based on TF-IDF to calculate the semantic similarity between the user query and the problems in the agriculture dataset. Moreover, it is required to classify the farmer complaint based on the problem category using SVM in Map/Reduce environment. This paper built a semantic model for the agricultural data to help farmers. As a result, significant effects of many important challenges and problems facing the agricultural sector are hoped to be minimized. The AgroSupportAnalytics system provides more accuracy than existing techniques using Precision, Recall, F1 score, and accuracy. It performs better accuracy 87% of LSA using TF-IDF with SVM classifier.

ACKNOWLEDGEMENTS

This work is supported by Newton Institutional grant ID 347762518, under Egypt Newton-Mosharafa Fund ID 30812 partnership. The grant is funded by 'UK Department for Business, Energy and Industrial Strategy' and 'Science and Technology Development Fund (STDF)' and delivered by the British Council.

REFERENCES

[1] M. E. Sykuta, "Big data in agriculture: property rights, privacy and competition in ag data services," *International Food and Agribusiness Management Review*, vol. 19, no. A, pp. 57–74, 2016.
[2] Y. Madani, M. Erritali, and J. Bengourram, "Sentiment analysis using semantic similarity and Hadoop MapReduce," *Knowledge and Information Systems*, vol. 59, no. 2, pp. 413–436, May 2019, doi: 10.1007/s10115-018-1212-z.
[3] B. J. Sun, Z. C. Liang, Q. T. Zeng, H. Zhao, W. J. Ni, and H. Duan, "Short text similarity computing method towards agriculture question and answering systems," *Advanced Materials Research*, vol. 756–759, pp. 1309–1313, Sep. 2013, doi: 10.4028/www.scientific.net/AMR.756-759.1309.
[4] G. Veeck, A. Veeck, and H. Yu, "Challenges of agriculture and food systems issues in China and the United States," *Geography and Sustainability*, vol. 1, no. 2, pp. 109–117, Jun. 2020, doi: 10.1016/j.geosus.2020.05.002.
[5] British Council and STDF, "Agro support analytics." https://agrosupportanalytics.com/ (accessed Sep. 01, 2021).
[6] The Ministry of Agriculture and Land Reclamation of Egypt, "Wheat cultivation: In the ancient lands (valley lands)." (in Arabic), Program Agriculture Research Center, http://www.vercon.sci.eg/indexUI/uploaded/wheatinoldsoil/wheatinoldsoil.htm#r1 (accessed Sep. 01, 2021).
[7] N. Grozev and R. Buyya, "Multi-cloud provisioning and load distribution for three-tier applications," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 9, no. 3, pp. 1–21, Oct. 2014, doi: 10.1145/2662112.
[8] E. Agirre *et al.*, "SemEval-2014 task 10: multilingual semantic textual similarity," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 81–91, doi: 10.3115/v1/S14-2010.
[9] V. N. Phu, V. T. N. Chau, and V. T. N. Tran, "SVM for English semantic classification in parallel environment," *International Journal of Speech Technology*, vol. 20, no. 3, pp. 487–508, Sep. 2017, doi: 10.1007/s10772-017-9421-5.
[10] A. Kashyap *et al.*, "Robust semantic text similarity using LSA, machine learning, and linguistic resources," *Language Resources and Evaluation*, vol. 50, no. 1, pp. 125–161, Mar. 2016, doi: 10.1007/s10579-015-9319-2.
[11] P. Achananuparp, X. Hu, and X. Shen, "The evaluation of sentence similarity measures," in *Data Warehousing and Knowledge Discovery*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 305–316.
[12] E. M. B. Nagoudi and D. Schwab, "Semantic similarity of Arabic sentences with word embeddings," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pp. 18–24.
[13] S. Alzahrani, "Cross-language semantic similarity of Arabic-English short phrases and sentences," *Journal of Computer Science*, vol. 12, no. 1, pp. 1–18, Jan. 2016, doi: 10.3844/jcssp.2016.1.18.
[14] M. Potthast, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Cross-language plagiarism detection," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 45–62, Mar. 2011, doi: 10.1007/s10579-009-9114-z.
[15] W. Wali, B. Gargouri, and A. Ben Hamadou, "Sentence similarity computation based on WordNet and VerbNet," *Computación y Sistemas*, vol. 21, no. 4, pp. 627–635, Jan. 2018, doi: 10.13053/cys-21-4-2853.
[16] L. Dai and H. Huang, "An English-Chinese cross-lingual word semantic similarity measure exploring attributes and relations," in *An English-Chinese Cross-lingual Word Semantic Similarity Measure Exploring Attributes and Relations*, 2011, pp. 467–476.
[17] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning, "Bilingual word embeddings for phrase-based machine translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1393–1398.
[18] T. K. Das and P. M. Kumar, "Big data analytics: a framework for unstructured data analysis," *International Journal of Engineering and Technology (IJET)*, vol. 5, no. 1, pp. 153–156, 2013.
[19] A. Momtaz and S. Amreen, "Detecting document similarity in large document collection using MapReduce and the Hadoop framework," University of Brac university, Bangladesh, 2012.
[20] J.-H. Lee, S. Park, C.-M. Ahn, and D. Kim, "Automatic generic document summarization based on non-negative matrix factorization," *Information Processing & Management*, vol. 45, no. 1, pp. 20–34, Jan. 2009, doi: 10.1016/j.ipm.2008.06.002.
[21] N. K. Nagwani, "Summarizing large text collection using topic modeling and clustering based on MapReduce framework," *Journal of Big Data*, vol. 2, no. 1, pp. 6–24, Dec. 2015, doi: 10.1186/s40537-015-0020-5.
[22] M. Birjali, A. Beni-Hssane, M. Erritali, and Y. Madani, "Information content measures of semantic similarity between documents based on Hadoop system," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Oct. 2016, pp. 187–192, doi: 10.1109/WINCOM.2016.7777212.
[23] R. Sahal, M. H. Khafagy, and F. A. Omara, "Big data multi-query optimisation with Apache Flink," *International Journal of Web Engineering and Technology*, vol. 13, no. 1, pp. 78–97, 2018, doi: 10.1504/IJWET.2018.092401.
[24] H. S. H. Abdel Azez, M. H. Khafagy, and F. A. Omara, "Optimizing join in HIVE star schema using key/facts indexing," *IETE Technical Review*, vol. 35, no. 2, pp. 132–144, Mar. 2018, doi: 10.1080/02564602.2016.1260498.
[25] K. Al-Sabahi, Z. Zhang, J. Long, and K. Alwesabi, "An enhanced latent semantic analysis approach for Arabic document summarization," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 8079–8094, Dec. 2018, doi: 10.1007/s13369-018-3286-z.
[26] M. Aiken, "An updated evaluation of Google Translate accuracy," *Studies in Linguistics and Literature*, vol. 3, no. 3, Jul. 2019, doi: 10.22158/sll.v3n3p253.
[27] H.-T. Duong and T.-A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Computational Social Networks*, vol. 8, no. 1, pp. 8–24, Dec. 2021, doi: 10.1186/s40649-020-00080-x.
[28] V. Mohan, J. Ilamathi, and Nithya, "Preprocessing techniques for text mining - an overview," *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2015.
[29] E. Rslan, M. H. Khafagy, K. Munir, and R. M.Badry, "English semantic similarity based on map reduce classification for agricultural complaints," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 12, pp. 235–242, 2021, doi: 10.14569/IJACSA.2021.0121231.
[30] R. A. Farouk, M. H. Khafagy, M. Ali, K. Munir, and R. M.Badry, "Arabic semantic similarity approach for farmers' complaints," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, pp. 348–358, 2021, doi:

10.14569/IJACSA.2021.0121038.
[31] G.-S. Victor, P. Antonia, and S. Spyros, "CSMR: A scalable algorithm for text clustering with cosine similarity and MapReduce," 2014, pp. 211–220.

BIOGRAPHIES OF AUTHORS

Esraa Rslan is currently a Lecturer Assistant at the Faculty of Computers and Information, Fayoum University. She received the B.S. degree from Faculty of Computers and Information Fayoum University, Egypt, in 2012 and the M.Sc. degree from Faculty of Computers and Information, Cairo University, Egypt, in 2018. She is currently pursuing the Ph.D. degree in Fayoum University. Her research interests include big data, NLP and Database. She can be contacted at email: esraa.rslan@fayoum.edu.eg.

Mohamed H. Khafagy is Head of Big Data Research Group at Fayoum University. Mohamed received his Ph.D. in computer science in 2009. He also works at Oracle Egypt as a consultant. Mohamed is the manager of the National Electronic Exam center in the Supreme Council of Universities. Mohamed worked as a postdoc in the DIMA group in Technique University Berlin in 2012. Mohamed established the first Big data Research group in Fayoum Univerisity in 2013. He has many publications in the area of Big Data, Cloud computing, and database. He can be contacted at email: mhk00@fayoum.edu.eg.

Mostafa Ali is a lecturer at Information Systems Department, Faculty of Computers and Information, Fayoum University, Egypt. Also, he is the vice manager of the National Electronic Exam center at the Supreme Council of Universities (SCU), Egypt. He got his B.Sc in 2006 and M.Sc in 2013 from Assiut University, Egypt. He got his Ph.D in 2020 from Computer Science department, Mysore University, India. His research interests are text mining, NLP, machine learning, big data and data science. He can be contacted at email: mam16@fayoum.edu.eg.

Kamran Munir is Associate Professor in Data Science, in Department of Computer Science and Creative Technologies, University of West of England, Bristol, UK. His current research projects are in areas of data science, big data and analytics, artificial intelligence and virtual reality mainly funded by the Innovate UK. He published over 60 research articles; and he is a regular PC member and editor of various conferences and journals. He can be contacted at email: Kamran2.Munir@uwe.ac.uk.

Rasha M. Badry is a lecturer at Information Systems Department, Faculty of Computers and Information, Fayoum University, Egypt. She is director of Crisis Management, Faculty of Computers and Information, Fayoum University. She also director of National Bank for Scientific Laboratories and Equipment, Supreme Council of Universities, Egypt. She got her Ph.D, M.Sc, and B.Sc in 2015, 2007, and 2003 from Faculty of Computers and Information, Helwan University, Egypt. Her research interests are NLP, Machine learning, Data Science. She can be contacted at email: rmb01@fayoum.edu.eg.