IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 14, 2021, accepted May 13, 2021, date of publication May 20, 2021, date of current version May 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3082403

ATTRACTIVE – An Auto-Updating Database for Experimental Protocols in Regenerative Medicine

YUAN-MAO HUNG[1], MONG-HSUN TSAI[2,3], LIANG-CHUAN LAI[2,4], AND ERIC Y. CHUANG[1,2]

[1]Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei 10617, Taiwan
[2]Bioinformatics and Biostatistics Core, Center of Genomic and Precision Medicine, National Taiwan University, Taipei 10617, Taiwan
[3]Institute of Biotechnology, National Taiwan University, Taipei 10617, Taiwan
[4]Graduate Institute of Physiology, College of Medicine, National Taiwan University, Taipei 10617, Taiwan

Corresponding authors: Liang-Chuan Lai (llai@ntu.edu.tw) and Eric Y. Chuang (chuangey@ntu.edu.tw)

ABSTRACT Many research articles are published on regenerative medicine every year. However, only a small proportion of these articles provide experimental methods on organ/tissue differentiation. Therefore, we developed a database – ATTRACTIVE (An auTo-updating daTabase foR experimentAl protoCols in regeneraTIVe mEdicine) – that collects journal articles with differentiation methods in regenerative medicine and updates itself automatically on a regular basis. Since the number of articles in regenerative medicine was insufficient and unbalanced, which limited the performance of the supervised learning algorithms, we proposed an algorithm that combines cosine similarity and linear discriminant functions to classify articles based on their titles and abstracts more efficiently. The results show that our proposed methods out-performed other machine learning algorithms such as k-nearest neighbors, support vector machine, and long short-term memory methods. The classification accuracy reached 94.62%, even with a small and unbalanced dataset. Lastly, we incorporated our classifier into the database for automatic updates. The database is available at http://attractive.cgm.ntu.edu.tw/.

INDEX TERMS Article classification, attractive, cosine similarity, regenerative medicine, text classification

I. INTRODUCTION

There are a considerable number of medical research articles published every year. Many of these articles are collected and stored in the National Center for Biotechnology Information (NCBI) - PubMed Central (PMC) [1] database, which allows researchers to access full-text articles for free. Researchers can also find these articles via the Google Scholar search engine. However, when it comes to searching for regenerative medicine research articles that include differentiation methods for specific tissues/organs, both PMC and Google Scholar give numerous results [2] (about 10,000∼200,000 results), and most of these either do not relate to organ/tissue differentiation topics directly or lack differentiation methods, such as using the broad term ''immunology''. This forms an obstacle for regenerative medicine researchers to access a comprehensive set of published differentiation methods.

A targeted literature database would make it more convenient for researchers to search organ/tissue differentiation methods. Such a database would need scheduled updates to make sure it included the newest published articles. However, to update the database manually would require considerable time and manpower. It is impractical to screen all the documents returned by Google Scholar and PMC, so human curation limits the possibility of collecting data comprehensively. Therefore, an auto-updating database would be a better way to solve this problem [3], [4]. The auto-updating database would need to collect documents, read the content, extract text features, and classify the text by itself on a regular basis. Although a crawler can collect articles from the web with text-mining techniques that read and extract text features, accurate classification of documents still remains problematic.

Some previous research used cosine similarity algorithms to cluster similar documents or text into the same category [5]–[7]. Although these cosine similarity algorithms showed good performance in clustering similar text, this method could be improved by adding efficient learning rules to train the classification model using deep learning algorithms [8]–[10], such as convolutional neural networks (CNN) or long-short term memory (LSTM). However, deep learning algorithms require a great deal of labeled data

The associate editor coordinating the review of this manuscript and approving it for publication was Vincenzo Conti.

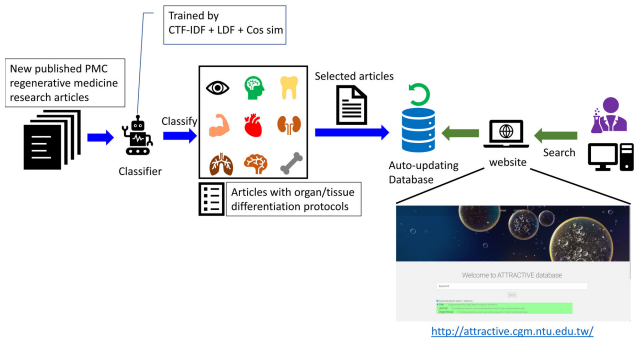

FIGURE 1. The overview of ATTRACTIVE database.

to train the classification model [11]–[13], because they take all the training patterns into consideration and modify the hyperplanes when separating different classes progressively.

Support vector machine (SVM) is another popular machine learning algorithm for text classification. Some previous research applied SVM algorithms to train classification models [14]–[16]. SVM requires less data than deep learning to reach a reasonable accuracy [17]. The reason for this is that the SVM algorithm only takes the boundary points of different classes into consideration. It then tries to separate the different classes by optimizing the distance of the hyperplane between the boundary points [18].

Both SVM and deep learning algorithms are supervised learning algorithms requiring a large amount of labeled data for training, and labeling training data can be a time-consuming process. Therefore, the purpose of this study was to develop an algorithm that can label published articles automatically by combining cosine similarity with linear discriminant learning rules to train classification models, using organ/tissue differentiation in regenerative medicine as a test case. In the process, we built an online regenerative medicine database containing differentiation protocols for different organs/tissues – ATTRACTIVE (An auTo-updating daTabase foR experimentAl protoCols in regeneraTIVe mEdicine). The trained model was embedded in the system so that the database can auto-update every six months. Fig. 1 shows the overview of the system. The URL of ATTRACTIVE is http://attractive.cgm.ntu.edu.tw/.

II. METHODS

Python 3.8.6 was used to develop our online database and algorithm. The source code of the proposed algorithm is available at: https://github.com/cil6758/ATTRACTIVE. The training algorithm included four parts – dataset collection, text preprocessing, feature extraction, and building learning models with a linear discriminant function (LDF) [19] (Fig. 2). Two organ/tissue classification models, one using titles and the other using abstracts, were built for comparison.

A. DATA COLLECTION

Since the amount of curated regenerative medicine articles in LifeMap Discovery database (https://discovery.lifemapsc.com/) [20] was limited and was not enough for machine learning algorithms to converge. In addition, some articles did not include organ/tissue keywords in the titles or abstracts, training the classifier to identify the indirect and related keywords is vital for correct classification. Therefore, we expanded the training set to enhance the terminology identification ability of the trained classifiers by the following method. Articles in the PMC database were collected to train organ/tissue classification models by E-utilities [21], the crawler provided by NCBI. The total article number in the training dataset was 17,239, as shown in Table 1.

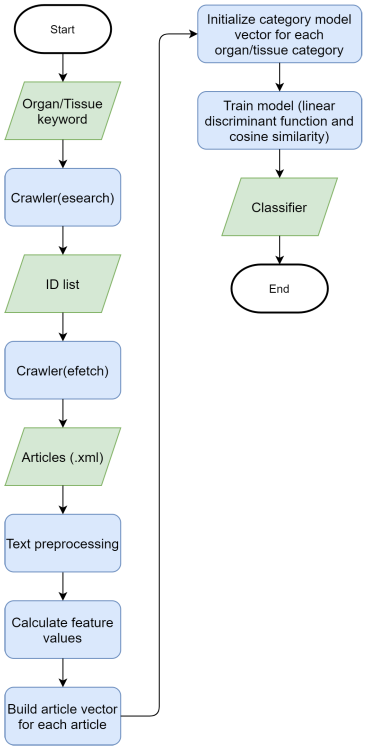

FIGURE 2. The workflow of training a classifier.

The keywords used to search the article were "[organ/tissue keyword]" + "stem cell". For instance, the terms, "kidney" + "stem cell", were used to search for articles related to kidney in the PMC database. In addition, the "[organ/tissue keyword]" also took formal terminologies into consideration for searching articles. We collected these terminologies from LifeMap Discovery [20]. All articles whose title mentioned any search keyword were downloaded and labeled as the same category under "[organ/tissue keyword]".

For articles in which organ/tissue keywords were not in the title, we trained our classification models to identify all the related keywords. For example, an article related to pancreatic stem cells might not have the keyword "pancreatic" in the title, but use "insulin-producing cells". The more titles and abstracts we collected, the more words our classification models could learn. By learning highly related keywords, the model could associate "pancreatic", "insulin", and "islet" with "pancreas". The collected training dataset is shown in Table 1.

TABLE 1. Training dataset.

Category	Article number
Adipose	1580
Astrocyte	99
Blood	4150
Bone	1289
Cartilage	577
Cornea	279
Dopaminergic	125
Ectoderm	29
Endoderm	97
Endothelium	351
Epidermis	174
Heart	1829
Kidney	303
Liver	639
Lung	395
Melanocyte	48
Mesoderm	41
Motor neuron	94
Muscle	534
Neuron	3360
Oligodendrocyte	40
Pancreas	317
Reproductive system	543
Retina	296
Schwann cell	17
Thyroid	33
Total	17239

B. TEXT PREPROCESSING

After papers were downloaded, Beautiful Soup 4 (version 4.8.2) [22] was used to extract the information from the downloaded articles. The extracted text was preprocessed by the following procedures: case folding, tokenizing, filtering, and stemming.

The case folding step transformed all uppercase letters to lowercase letters. In the tokenizing step, sentences were split into words, and punctuation and numerals were removed. In the filtering step, each word was mapped to the NLTK dictionary [23] and stop words (e.g., ''the'', ''is'', ''and'', etc.) were removed. In the stemming step, we stemmed the extracted root of each word, because a word may have different spellings in different parts of speech (noun, verb, adjective, adverb, etc.). Stemming text words can avoid regarding variants of a word in different parts of speech as completely different features.

C. FEATURE EXTRACTION

After the above steps of text preprocessing, each remaining word was considered a feature. We used a category-based term frequency – inverse document frequency (CTF-IDF) method to calculate the feature values.

1) TERM FREQUENCY (TF)

Equation (1) is the formula used to calculate the term frequency of a word in a document:

$$TF(t) = \frac{f_{t,d}}{T_d}. \tag{1}$$

where $TF(t)$ is the frequency of term t in a document d. $f_{t,d}$ is the occurrence time of term t in document d. T_d is the total word number of document d.

2) TERM FREQUENCY –inverse DOCUMENT FREQUENCY (TF-IDF)

Equation (2) is the formula for inverse document frequency:

$$IDF(t) = \log\left(\frac{D_{total}}{D_{term}}\right) \tag{2}$$

where $IDF(t)$ is the inverse document frequency value of term t. D_{total} is the total document number in the dataset. D_{term} is the number of documents where term t appears.

In (2), the IDF value becomes zero when D_{term} equals D_{total}, which means term t appears in all the documents. This indicates that term t cannot be a feature used to distinguish documents efficiently. Therefore, the IDF weight value becomes zero.

Next, we can calculate the TF-IDF value for each term by multiplying TF by IDF value as shown in (3):

$$TF\text{-}IDF(t) = TF(t) \times IDF(t) \tag{3}$$

If a term appears frequently in a single document but rarely appears in other documents, it has high TF and IDF values, indicating that the term is a good specific feature of the article.

3) CATEGORY-BASED TF-IDF (CTF-IDF)

In order to identify the representative words for each organ/tissue category, we improved the TF-IDF value by calculating the category-based TF-IDF (CTF-IDF), as shown in (4):

$$CTF\text{-}IDF(t) = TF(t) \times IDF(t) \times \log\left(\frac{C_{total}}{C_{term}}\right) \tag{4}$$

where $CTF\text{-}IDF(t)$ is the category-based TF-IDF value of term t. $TF(t)$ is the term frequency of term t. $IDF(t)$ is the inverse document frequency value of term t. C_{total} is the total number of categories. C_{term} is the number of categories in which term t appears.

In (4), the log value becomes zero when C_{term} equals C_{total}, which means term t appears in all the categories. This also indicates that term t cannot be used to distinguish the documents in different categories efficiently.

4) BUILDING THE ORGAN/TISSUE STATIC MODEL

Each article was assigned to the same category with the searching organ/tissue keyword, and every word in each article was assigned its own TF, TF-IDF, and CTF-IDF values. Hereafter we call these values ''feature values''. For an article, the feature values of the terms can form a vector of the article as shown in Fig. 3a. The sum of all the article vectors in a specific organ/tissue category can further form a category vector. The category vector is the static model for each specific organ/tissue. Fig. 3b illustrates the concept, where articles 1 to 5 belong to the category ''pancreas'' while articles 6 to 9 belong to the category ''heart''. Then, the static

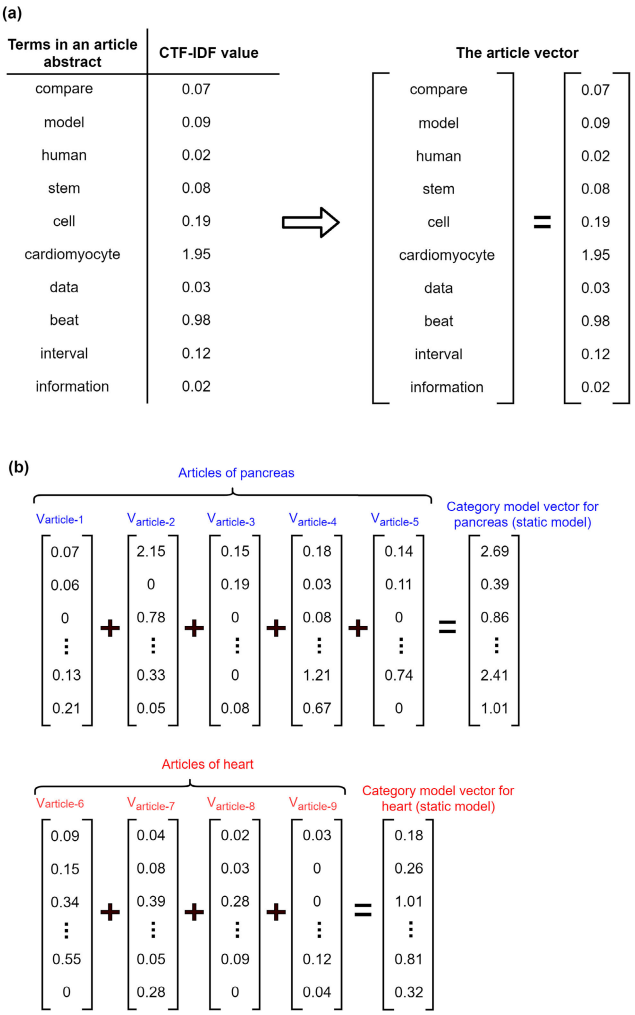

FIGURE 3. **(a) Example of using CTF-IDF feature values to form an article vector from an article abstract. (b) Example of forming category vectors (static models) for each organ/tissue category.**

category model for ''pancreas'' is the sum of the vectors for articles 1 to 5, while the static category model for ''heart'' is the sum of the vectors for articles 6 to 9. Equation (5) shows how to form the static model for each category:

$$\vec{M_i} = \sum_{j=1}^{n_i} \vec{V_a}(j) \tag{5}$$

where $\vec{M_i}$ is the i-th organ/tissue static category model vector. The $\vec{M_i}$ category includes n_i articles. $\vec{V_a}(j)$ is the j-th article vector which belongs to category $\vec{M_i}$.

5) CLASSIFYING ARTICLES BASED ON COSINE SIMILARITY

Since each article has its own article vector and the static model for each organ/tissue category is also a vector, to evaluate the similarity between article and category, we can calculate the cosine similarity [7] between an article and the organ/tissue static models and assign the article to the category which has the highest similarity score. Equation (6) shows the formula of cosine similarity:

$$Sim(\vec{M}, \vec{V}_a) = \frac{\vec{M} \cdot \vec{V}_a}{\|\vec{M}\| \|\vec{V}_a\|} = \frac{\sum_{k=1}^{n} (\vec{M}_k \times \vec{V}_{ak})}{\sqrt{\sum_{k=1}^{n} (\vec{M}_k)^2} \times \sqrt{\sum_{k=1}^{n} (\vec{V}_{ak})^2}} \tag{6}$$

where $\vec{M}$ is one of the organ/tissue static model vectors, $\vec{V}_a$ is the article to be classified, and $\vec{M}_k$ and $\vec{V}_{ak}$ are the k-th component of vectors $\vec{M}$ and $\vec{V}_a$, respectively.

6) ADDING THE LEARNING RULE – LDF

To improve the static organ/tissue category model, LDFs [19] were used. First, one article was chosen from the training dataset and cosine similarity between the selected article and every organ/tissue model was calculated, and the article was assigned to the category with the highest score. Next, the algorithm checked whether the assigned category was the same as the label (each article from the training dataset was initially labeled). If the assignment was not consistent with the label, the wrongly assigned article vector would be subtracted from the wrong organ/tissue model vector and the correctly assigned article vector would be added to the correct organ/tissue model, as shown in (7) and (8).

$$\vec{M}'_{wrong} = \vec{M}_{wrong} - \vec{V}_a \tag{7}$$

$$\vec{M}'_{correct} = \vec{M}_{correct} + \vec{V}_a \tag{8}$$

where $\vec{M}_{wrong}$ and $\vec{M}'_{wrong}$ are the original and improved wrongly classified organ/tissue category models, respectively. $\vec{M}_{correct}$ and $\vec{M}'_{correct}$ are the original and improved correct category models, respectively. $\vec{V}_a$ is the article vector.

Equations (7) and (8) can enhance the classification ability of the model by highlighting the feature values (TF, TF-IDF, CTF-IDF) of the critical terms in the model vector. For instance, in the ''kidney'' category, critical terms such as ''kidney'' and ''nephron'' have much higher weight values than other regular words.

This algorithm used all the training articles to improve the organ/tissue model for each round. If there was one article that was assigned to the wrong category, the algorithm would proceed another round until all articles were correctively assigned to the right categories. Note that some of the elements in the category vectors would be less than zero in the LDF method. When an article included no keywords related to regenerative medicine, the article was assigned to the ''unclassified'' category.

7) EXPERIMENTAL SETUP

Our experiments used five different training methods to measure the performance of article classification: static modeling, cosine similarity + LDF (Cos + LDF), SVM, K-nearest neighborhood (KNN) [24], and long short-term memory (LSTM) [25].

For the static modeling and Cos + LDF methods to classify an unlabeled article, the TF values for each term of the article were extracted, and cosine similarity between the article and static model vectors of each category was calculated. The article was assigned to the category with the highest similarity score. For the Cos + LDF method, if an article had cosine similarity score < 0 to all the organ/tissue category vectors, the article was assigned to the ''unclassified'' category.

The scikit-learn package [26] was used to train the SVM model. We applied the ''linear'' kernel because it provides better text-classification performance, according to Kalcheva *et al.* [14]. Regularization parameter C was set to 10 and the ''class_weight'' parameter was set to ''balanced'' due to the unbalanced characteristics of our training data.

The sickit-learn package was also used to train the KNN model. The k-value was set to 5 because it could provide better classification performance according to [27]–[29].

Tensorflow [30] and Keras [31] were used to train the LSTM model. A two-tier stacked model [10] was chosen and the hidden units were set to 256. The loss function was ''categorical_crossentropy''.

8) VALIDATION

The articles from LifeMap Discovery [20] were used as our validation dataset, because these articles were curated by human professionals. The validation dataset contained 260 articles with 26 organ/tissue categories: pancreas, kidney, epidermis, cornea, lung, muscle, Schwann cell, dopaminergic neuron, heart, motor neuron, neuron, melanocyte, thyroid, mesoderm, liver, endoderm, ectoderm, bone, cartilage, astrocytes, oligodendrocyte, reproductive system, retina, blood, adipose and endothelium (Table 2).

Notice that there are some overarching relationships between these categories. Endoderm includes liver, lung, pancreas, and thyroid. Mesoderm includes adipose, blood, bone, cartilage, endothelium, heart, kidney, reproductive system, and muscle. Ectoderm includes neuron, astrocyte, dopaminergic neuron, motor neuron, oligodendrocyte, epidermis, cornea, retina, melanocyte, and Schwann cell. The articles discussed neuron differentiation topic but not mainly for astrocyte, oligodendrocyte, dopaminergic neuron, and motor neurons were assigned to the ''neuron'' category. The reason to regard endoderm, mesoderm, and ectoderm as separate categories and allow them to have their own model vectors is that some of the articles did not have explicit organ/tissue keywords in their titles, or their abstracts included only a few organ/tissue keywords. This could result in failed classification of the articles. Allowing endoderm, mesoderm, and ectoderm to have their own model vectors increased the possibility of capturing these inexplicit articles when comparing the similarity scores between different categories. In the validation procedure, if the classification model could classify an article to the correct organ/tissue or corresponding germ layer, we regarded it as a correct classification.

There are several ways to evaluate text classification results, e.g., accuracy, precision, recall, and F-measure.

TABLE 2. Validation dataset.

Category	Article number
Adipose,mesoderm	7
Astrocyte,ectoderm	3
Blood,mesoderm	10
Bone,mesoderm	12
Cartilage,mesoderm	11
Cornea,ectoderm	5
Dopaminergic,ectoderm	20
Endoderm	3
Endothelium,mesoderm	6
Epidermis,ectoderm	4
Heart,mesoderm	42
Kidney,mesoderm	14
Liver,endoderm	8
Lung,endoderm	4
Melanocyte,ectoderm	1
Motor neuron,ectoderm	27
Muscle,mesoderm	17
Neuron,ectoderm	12
Oligodendrocyte,ectoderm	5
Pancreas,endoderm	23
Reproductive system,mesoderm	6
Retina,ectoderm	11
Schwann cell,ectoderm	8
Thyroid,endoderm	1
Total	260

These metrics can be calculated from a confusion matrix and the formulas are listed in (9)-(12):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

where TP, TN, FP, and FN mean true positive, true negative, false positive, and false negative, respectively. Accuracy is equivalent to the ratio of correctly assigned articles divided by the total article number. Precision indicates, when the classifier assigns a number of articles to a specific category, the proportion of assigned articles that truly belong to the category. Recall indicates the proportion of articles in a specific category that is assigned correctly. F-measure takes both precision and recall into consideration, which provides more objective information to evaluate the results.

9) DATABASE APPLICATION

To assist users with literature searches related to organ/tissue experimental methods, we developed an online database. The classifier was integrated into the database so that the system could assign the collected articles to corresponding organ/tissue categories. The database used the crawler to download the published articles from the NCBI PMC database and will update its content every six months using the Linux ''crontab'' command.

The database updating workflow is shown in Fig. 4. After downloading articles from the PMC database, the system

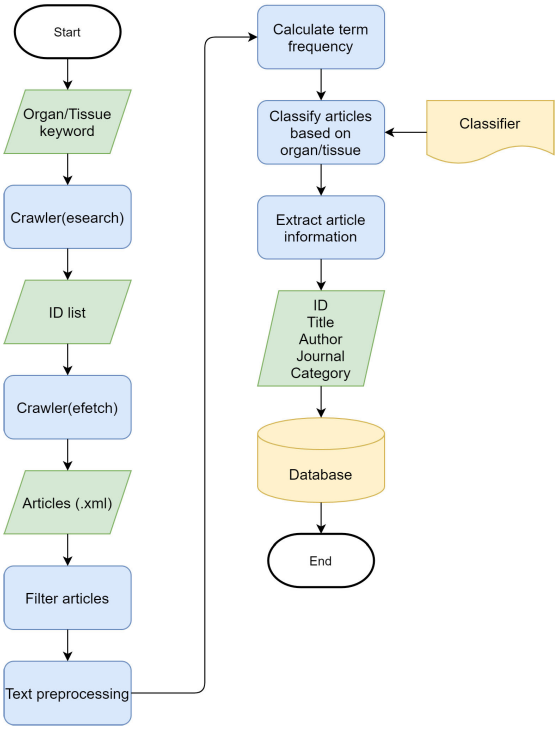

FIGURE 4. Integration of the classifier into the auto-updating database.

TABLE 3. Feature value metrics comparison.

	TF (%)	TF-IDF (%)	CTF-IDF (%)
Title	69.23	80.00	90.00
Abstract	80.38	90.77	93.85

conservatively selected the articles that explicitly included ''differentiation'' and ''stem cell'' keywords in the titles. Then, the algorithm took the articles that included experimental protocols into consideration. Next, text preprocessing is applied to the selected articles and the system calculates the term frequency and builds the term vector for each article. After that, the system calculates cosine similarity between the article term vector and the organ/tissue model vectors stored in the classifiers. An article is assigned to the category with the highest similarity score. Finally, the system extracts information on each article, such as ''ID'', ''title'', ''author'', ''journal name'', and ''organ/tissue category'', and stores the information in the database. The web page of the database is shown in Fig. 5.

D. RESULTS

1) COMPARISON OF DIFFERENT FEATURE VALUE METRICS

In order to build static organ/tissue category models for classification, the accuracy of using different feature value metrics was compared (Table 3). The TF-IDF metric performed better than the TF metric, as expected, and the CTF-IDF performed best among the three different metrics (title 90% and abstract 93.85%). This indicated that CTF-IDF had good ability to distinguish articles from different categories.

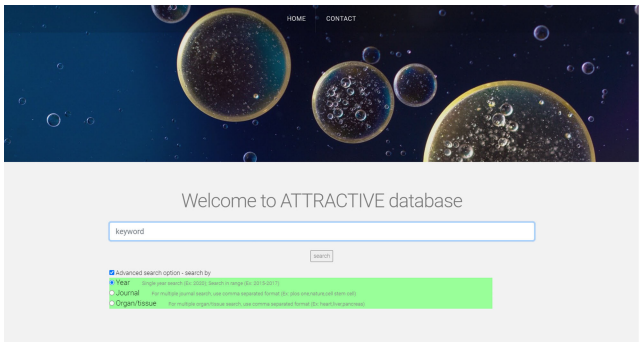

FIGURE 5. The web page of the regenerative medicine database (ATTRACTIVE).

TABLE 4. Accuracy of different training algorithms.

Model	Target	Method	Accuracy (%)
Title	Title	Static model	90.00
		Cos+LDF	90.77
		SVM	28.84
		KNN(k=5)	29.23
		LSTM	72.31
Title	Abstract	Static model	93.46
		Cos+LDF	94.23
		SVM	28.85
		KNN(k=5)	29.23
		LSTM	67.31
Abstract	Title	Static model	92.69
		Cos+LDF	91.15
		SVM	15.38
		KNN(k=5)	14.61
		LSTM	73.46
Abstract	Abstract	Static model	93.85
		Cos+LDF	94.62
		SVM	22.31
		KNN(k=5)	15.77
		LSTM	78.08

2) CLASSIFICATION RESULTS

The accuracy of different algorithms used for classification was compared (Table 4). The CTF-IDF metric was used to build organ/tissue model vectors and train the classification models. There were two types of classification models: title and abstract. All the article titles and abstracts in the training dataset were used to train the title and abstract classification models by different algorithms. Each of these two models were used to classify articles from the validation dataset. In Table 4, the ''Model'' column indicates the applied model type, the ''Target'' column indicates the classification target (title or abstract), the ''Method'' column indicates the algorithms used to build the classification model, and the ''Accuracy'' column shows the accuracy of each model trained by different algorithms.

The results indicate that the static organ/tissue model and Cos+LDF algorithm performed better than the other three algorithms, i.e., SVM, KNN, and LSTM. Generally, the Cos+LDF algorithm had better performance than the static organ/tissue model, but when we used the abstract model to classify article titles, the Cos+LDF algorithm

TABLE 5. Precision, recall, and F-measure for abstract model-article abstract classification.

	Precision	Recall	F-measure
Adipose	0.60	0.86	0.71
Astrocyte	0.67	0.67	0.67
Blood	1.00	0.80	0.89
Bone	1.00	0.50	0.67
Cartilage	1.00	0.88	0.93
Cornea	1.00	1.00	1.00
Dopaminergic neuron	0.94	0.80	0.87
Endothelium	0.86	0.67	0.75
Epidermis	1.00	1.00	1.00
Heart	1.00	0.98	0.99
Kidney	1.00	1.00	1.00
Liver	0.88	0.88	0.88
Lung	1.00	0.80	0.89
Melanocyte	1.00	1.00	1.00
Motor neuron	1.00	0.85	0.92
Muscle	0.93	0.82	0.88
Neuron	0.77	0.83	0.80
Oligodendrocyte	1.00	1.00	1.00
Pancreas	1.00	0.87	0.93
Reproductive system	1.00	1.00	1.00
Retina	1.00	1.00	1.00
Schwann cell	1.00	0.63	0.77
Thyroid	1.00	1.00	1.00

had a little lower accuracy (about 1.54%) than the static organ/tissue model. Among all the model-target combinations and different algorithms, using the abstract model to classify article abstracts provided the best performance (94.62%). The SVM and KNN methods had poor classification accuracy (<30%). For the LSTM learning method, the accuracy ranged from 67.31% to 78.08%. This indicated that LSTM could learn features from an unbalanced dataset better than SVM and KNN.

Since using abstract model to classify articles based on their abstracts by the Cos+LDF algorithm provided the best performance, the precision, recall, and F-measure of this model were compared (Table 5). The germ layer categories ''ectoderm'', ''mesoderm'', and ''endoderm'' were removed from Table 5, because these categories were designed to capture the articles that lack organ/tissue keywords in their titles or abstracts. If an article shows obvious organ/tissue keywords in its title or abstract, the classifier assigns the article to the explicit organ/tissue category, instead of only the germ layer name. Precision ranged from 0.6 to 1, recall ranged from 0.5 to 1, and F-measure ranged from 0.67 to 1.

Lastly, we developed a database of articles on regenerative medicine by applying the Cos+LDF abstract classification model (Fig. 5). The articles in this database will be updated automatically every six months. Users can search for protocols related to regenerative medicine directly based on keywords. In addition, users can search articles based on publication years, journal, or organ/tissue type in the ''advanced search'' mode. This mode allows users to search the database for differentiation methods published during a specific time range, articles published in journals highly focused on regenerative medicine research, or all articles related to specific organs or tissues.

E. DISCUSSION

In this work, we developed an algorithm that combined cosine similarity and LDFs to classify journal articles efficiently. A category-based TF-IDF feature value metric was used to build static model vectors for each category; additionally, LDFs was applied to further improve the category vectors and cosine similarity for classifying articles. Our method could work well even in limited data size and unbalanced data distribution. Furthermore, we integrated our classification model into a database whose contents will be updated automatically.

The feature value metric experiments showed that CTF-IDF has the best performance. CTF-IDF was able to distinguish and highlight the feature terms for each category. When there were multiple categories, CTF-IDF could enhance the performance of text classification. Therefore, CTF-IDF was applied in the subsequent experiments.

For classification accuracy, the static model and Cos + LDF showed much better performance than SVM and KNN, indicating that the static model and Cos + LDF methods could identify the keywords for each category. For the static model method, the weight of each keyword was emphasized by summing up the CTF-IDF term vectors of the articles belonging to the corresponding category. If an unlabeled article title or abstract included keywords belonging to a specific organ/tissue model, it received a high similarity score due to the large weight of the keywords. For the Cos + LDF training method, an LDF was applied to further improve the static model by enhancing the weight of the keywords iteratively. This also improves the ability to distinguish articles between different categories.

In our experiments, SVM had low accuracy. There are several possible reasons for this result. The first is the unbalanced training dataset. Unbalanced training causes overfitting problems in the learning procedure and results in incorrect classification [32]. Insufficient training data size is a second possible reason. As mentioned before, machine learning algorithms usually require a large amount of training data to learn features. In our experiments, some of the organ/tissue categories contained an insufficient number of articles for SVM to learn the features correctly. Most of the categories included fewer than 1,000 training articles. The third reason might be the curse of dimensionality as indicated in [33]. In our experiments, there were 26 categories for classification and a large number of term features to learn. That meant that much more training data was needed to overcome the high dimensionality problem of SVM.

KNN has the same problems as SVM. The unbalanced training data caused serious bias in the classification results [29]. Most of the articles were assigned to categories that possessed a large amount of training data, such as the ''blood'' and ''neuron'' categories.

However, LSTM, one of the deep learning algorithms, was not affected much by the unbalanced dataset and had an accuracy of 67.31-78.08%. The reason why LSTM achieved much better performance than SVM and KNN might be that LSTM analyzed the training articles based on full text instead

of keywords and term frequency metrics (TF, TF-IDF, and CTF-IDF). Full text analysis allows LSTM to possess more information than SVM and KNN. However, the training data size was still not enough and limited the performance of LSTM.

Since the dataset is unbalanced, the F-measure provides a more accurate measurement than precision or recall alone, as it takes both terms into consideration [34]. The ''astrocyte'' and ''bone'' categories had the lowest F-measures of all the categories (both 0.67). We observed that both of these categories correlated to the keywords of multiple categories in the text. ''Astrocyte'' frequently correlated to ''oligodendrocyte'' and ''motor neuron'', while ''bone'' correlated to many mesoderm organ/tissue categories (data not shown). There was a term, ''bone marrow stem cell'', which appeared in many article titles and abstracts and which caused CTF-IDF to greatly decrease the weight of the term ''bone''. However, some of the articles only used ''bone'' instead of more specific terms such as ''osteoblast'' or ''osteocyte'' in their titles and abstracts. This situation made it harder for the classifier to classify the articles in the ''bone'' category correctly due to the low weight of the organ/tissue keyword. Therefore, the ''bone'' category had a lower recall value (0.5) and F-measure (0.67). This could also be the reason why the ''astrocyte'' category had a lower F-measure.

Our method could be applied in situations where there is limited data size, unbalanced data distribution, and many categories for classification. This situation is common in the microbial taxonomy assignment problem in the field of molecular biology, since there is a large number of species but a limited number of curated conserved genes. Therefore, our proposed algorithm might also be suitable for taxonomy assignment problems.

One limitation should be noted regarding our proposed method. As mentioned before, we used CTF-IDF as the feature value metric to enhance the distinguishing ability of the classification model. However, if there were only a few categories (e.g., two or three) for classification, it is highly possible that the keywords would appear in all the categories. This situation would make the critical feature terms of each category close or equal to zero, causing the model to lose its classification ability. Therefore, we emphasize here that the CTF-IDF metric provides better performance in the situation where many categories are available for classification. If only a few categories are available for classification, the TF-IDF metric is suggested.

REFERENCES

[1] R. J. Roberts, ''PubMed central: The GenBank of the published literature,'' *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 381–382, Jan. 2001.

[2] P. Jacsó, ''Google scholar revisited,'' *Online Inf. Rev.*, vol. 32, no. 1, pp. 102–114, Feb. 2008.

[3] W. Lam and K. S. Ho, ''FIDS: An intelligent financial Web news articles digest system,'' *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 31, no. 6, pp. 753–762, Nov. 2001.

[4] C.-C. Chen and C.-L. Ho, ''StemTextSearch: Stem cell gene database with evidence from abstracts,'' *J. Biomed. Informat.*, vol. 69, pp. 150–159, May 2017.

[5] P. Y. Ristanti, A. P. Wibawa, and U. Pujianto, ''Cosine similarity for title and abstract of economic journal classification,'' in *Proc. 5th Int. Conf. Sci. Inf. Technol. (ICSITech)*, Oct. 2019, pp. 123–127.

[6] P. D. Nurfadila, A. P. Wibawa, I. A. E. Zaeni, and A. Nafalski, ''Journal classification using cosine similarity method on title and abstract with frequency-based stopword removal,'' *Int. J. Artif. Intell. Res.*, vol. 3, no. 2, pp. 28–37, Jul. 2019.

[7] K. Rinartha and L. G. S. Kartika, ''Scientific article clustering using string similarity concept,'' in *Proc. 1st Int. Conf. Cybern. Intell. Syst. (ICORIS)*, Aug. 2019, pp. 13–17.

[8] J. Wang, Y. Li, J. Shan, J. Bao, C. Zong, and L. Zhao, ''Large-scale text classification using scope-based convolutional neural network: A deep learning approach,'' *IEEE Access*, vol. 7, pp. 171548–171558, 2019.

[9] C. Li, G. Zhan, and Z. Li, ''News text classification based on improved Bi-LSTM-CNN,'' in *Proc. 9th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Oct. 2018, pp. 890–893.

[10] Y. Luan and S. Lin, ''Research on text classification based on CNN and LSTM,'' in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Mar. 2019, pp. 352–355.

[11] R. Vinayakumar, K. P. Soman, and P. Poornachandran, ''Evaluation of recurrent neural network and its variants for intrusion detection system (IDS),'' *Int. J. Inf. Syst. Model. Des.*, vol. 8, no. 3, pp. 43–63, Jul. 2017.

[12] M. I. Razzak, S. Naz, and A. Zaib, ''Deep learning for medical image processing: Overview, challenges and the future,'' in *Classification in BioApps: Automation of Decision Making*. Cham, Switzerland: Springer, 2018, pp. 323–350.

[13] R. Vinayakumar, K. P. Soman, and P. Poornachandran, ''Applying convolutional neural network for network intrusion detection,'' in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228.

[14] N. Kalcheva, M. Karova, and I. Penev, ''Comparison of the accuracy of SVM kemel functions in text classification,'' in *Proc. Int. Conf. Biomed. Innov. Appl. (BIA)*, Sep. 2020, pp. 141–145.

[15] J. Ouyang, ''Research on english text information filtering algorithm based on SVM,'' in *Proc. IEEE Int. Conf. Power, Intell. Comput. Syst. (ICPICS)*, Jul. 2020, pp. 1001–1004.

[16] Z. Liu, X. Lv, K. Liu, and S. Shi, ''Study on SVM compared with the other text classification methods,'' in *Proc. 2nd Int. Workshop Educ. Technol Comput. Sci.*, vol. 1, 2010, pp. 219–222.

[17] P. Liu, K.-K.-R. Choo, L. Wang, and F. Huang, ''SVM or deep learning? A comparative study on remote sensing image classification,'' *Soft Comput.*, vol. 21, no. 23, pp. 7053–7065, Dec. 2017.

[18] W. S. Noble, ''What is a support vector machine?'' *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, ''Linear discriminant functions,'' in *Pattern Classification*. New York, NY, USA: Wiley, 2001, pp. 215–281.

[20] R. Edgar, Y. Mazor, A. Rinon, J. Blumenthal, Y. Golan, E. Buzhor, I. Livnat, S. Ben-Ari, I. Lieder, A. Shitrit, Y. Gilboa, A. Ben-Yehudah, O. Edri, N. Shraga, Y. Bogoch, L. Leshansky, S. Aharoni, M. D. West, D. Warshawsky, and R. Shtrichman, ''LifeMap discovery: The embryonic development, stem cells, and regenerative medicine research portal,'' *PLoS ONE*, vol. 8, no. 7, 2013, Art. no. e66629.

[21] G. Mias, ''Databases: E-utilities and UCSC genome browser,'' in *Mathematica for Bioinformatics: A Wolfram Language Approach to Omics*. Cham, Switzerland: Springer, 2018, pp. 133–170.

[22] G. L. Hajba, ''Using beautiful soup,'' in *Website Scraping With Python: Using BeautifulSoup and Scrapy*. Berkeley, CA, USA: Apress, 2018, pp. 41–96.

[23] S. Bird, ''NLTK: The natural language toolkit,'' in *Proc. COLING/ACL Interact. Presentation Sessions*, 2006, pp. 69–72.

[24] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, ''A novel kNN algorithm with data-driven K parameter computation,'' *Pattern Recognit. Lett.*, vol. 109, pp. 44–54, Jul. 2018.

[25] A. Graves, ''Long short-term memory,'' in *Supervised Sequence Labelling With Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 37–45.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[27] M. J. Islam, Q. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed, ''Investigating the performance of naive-Bayes classifiers and K-nearest neighbor classifiers,'' in *Proc. Int. Conf. Converg. Inf. Technol. (ICCIT)*, 2007, pp. 1541–1546.

[28] G. Batista and D. F. Silva, "How K-nearest neighbor parameters affect its performance," in *Proc. Argentine Symp. Artif. Intell.*, 2009, pp. 1–12.
[29] J. E. Goin, "Classification bias of the K-nearest neighbor algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 3, pp. 379–381, May 1984.
[30] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th Symp. Oper. Syst. Design Implement.*, 2016, pp. 265–283.
[31] J. Moolayil, "An introduction to deep learning and Keras," in *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning With Python*. Berkeley, CA, USA: Apress, 2019, pp. 1–16.
[32] H. N. Castro, L. G. Abril, and C. A. Bahón, "A post-processing strategy for SVM learning from unbalanced data," in *Proc. 19th Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, 2011, pp. 195–200.
[33] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, 2005, pp. 758–770.
[34] A. Abid, W. Ali, M. S. Farooq, U. Farooq, N. S. Khan, and K. Abid, "Semi-automatic classification and duplicate detection from human loss news corpus," *IEEE Access*, vol. 8, pp. 97737–97747, 2020.

YUAN-MAO HUNG received the B.S. degree from the National Central University of Electrical Engineering, Zhongli, Taiwan, in 2012, and the M.S. degree from the National Chiao Tung University of Communication Engineering, Hsinchu, Taiwan, in 2015. He is currently pursuing the Ph.D. degree with the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan. His research interests include metagenomics and computer aided engineering.

MONG-HSUN TSAI received the B.S. degree in zoology from National Taiwan University, Taipei, Taiwan, in 1993, the M.S. degree in radiation biology from National Tsing Hua University, Hsinchu, Taiwan, in 1995, and the Ph.D. degree in public health from National Yang-Ming University, Taipei, Taiwan, in 2001. He is currently working as a Professor and the Director with the Institute of Biotechnology, National Taiwan University. His research interests include bioinformatics, microarray, cancer biology, and radiation biology.

LIANG-CHUAN LAI received the M.S. and Ph.D. degrees in molecular and integrative physiology from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2001 and 2005, respectively. He is currently working as a Professor with the Department of Physiology, National Taiwan University, Taipei, Taiwan. His research interests include using genomic approaches, i.e. microarrays and next generation sequencing, to explore the molecular mechanism of carcinogenesis.

ERIC Y. CHUANG received the Ph.D. degree in cancer biology with toxicology and molecular genetics as two sub-specialties from Harvard University, in 1997, and the Executive M.B.A. degree in international business management from National Taiwan University (NTU), in June 2017. After graduation, he stayed at Harvard as a Postdoctoral Fellow for a period of one year. He then joined the Radiation Biology Branch of National Cancer Institute (NCI), National Institutes of Health (NIH), as an IRTA Fellow to study radiogenomics in Bethesda, MD, USA. He became the Head of the Microarray Laboratory for Radiation Oncology Sciences Program, NCI. After working with the NIH for several years, he took a faculty position with NTU. He joined the Radiation Research Program of Division of Cancer Treatment and Diagnosis, NCI, as a Program Director, in 2009, to oversee a portfolio of NIH grants that included radiation-induced signaling pathways, molecular mechanisms and normal tissue injuries as well as radiation related genomic studies. He returned to NTU, in 2011. From 2012 to 2018, he was working as the Director of the Graduate Institute of Biomedical Electronics and Bioinformatics (BEBI). He is currently a Professor with BEBI, NTU. He is also working as the Dean of the College of Biomedical Engineering, China Medical University. Being an expert in genomic technologies, bioinformatics, cancer, radiation biology and oncology, biomedical engineering, and precision medicine, he has published more than 127 peer-reviewed articles in related fields. His Ph.D. thesis was to study radiation-induced mutagenesis in human cells. He has been serving as an editorial board member of Scientific Reports, and the Editor-in-Chief of *Translational Cancer Research*.