Preface

Special Issue on Advances in Intelligent Systems

This special issue presents the work of international research teams on new approaches to building intelligent systems.

The first two papers present new similarity measures for item-based neighborhood collaborative filtering and a genetic algorithm of green cellular network optimization.

A group of papers considers hybrid fuzzy models and systems: a fuzzy analog of the Central Limit Theorem, a new method for training fuzzy models based on the fuzzy Bayesian approach, an input-weighted multi-objective evolutionary fuzzy classifier, and fuzzy methods for comparing project situations and selecting precedent decisions.

The issue also contains original papers related to the rapidly developed last years advanced methods of Natural Language Processing, including Sentiment Analysis, Machine Learning, Deep Learning, Transfer Learning, and other methods. The papers explore the problems of concreteness rating estimation of English words, automatic language identification in mixed language texts, automatic abusive language detection, and automatic detection of opposition relations in legal texts.

The issue ends with two surveys. One survey reviews the publicly available datasets of fake news in low/medium-resourced Asian and European languages summarizing the methods used to evaluate the classifiers in identifying fake news. Another survey reviews the machine learning methods applied to physical sciences.

The Guest Editors thank Prof. Imre J. Rudas for supporting the publication of this Special Issue.

Ildar Batyrshin, Fernando Gomide, Vladik Kreinovich and Shahnaz Shahbazova

Guest Editors

New Similarity Measures for Item-based Neighborhood Collaborative Filtering

Eliuth E. López-García, Ildar Batyrshin, Grigori Sidorov

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Str. Juan de Dios Batiz, P.C. 07320, Mexico City, Mexico
eelopezg1700@alumno.ipn.mx, {batyr1, sidorov}@cic.ipn.mx

Abstract: Similarity measures play an important role in many areas to solve a wide variety of problems. In computer science, these measures are used in decision making, information retrieval, data mining, machine learning, and recommender systems. The recommender systems are tools that have proven their utility in filtering large amounts of information and giving recommendations useful for users. Neighborhood collaborative filtering is the most common recommender system approach implemented by cutting-edge companies. A key element of this approach is the similarity measure, which is used to find neighbors with similar tastes to provide recommendations that satisfy users' needs. A drawback of this approach is the lack of user's information to generate proper recommendations. For this reason, it is important to design new similarity measures that can find the most relevant neighbors to generate more accurate recommendations for users with little information about them. This paper designs two new similarity measures that can generate good recommendations with little information about users. These similarity measures have been tested using MovieLens datasets and different rating prediction methods, and they have shown a good performance in comparison with other similarity measures designed to address the recommendation problem.

Keywords: rating scale; recommender systems; collaborative filtering; neighborhood; item-based; similarity measure; similarity; cold-start

1 Introduction

Recommender systems (RS) are software tools and techniques providing suggestions for items (e.g., movies, songs, books, applications, websites, travel destinations, and e-learning material) to be of use to a user [1]. These systems are created to tackle the need to filter the amount of information generated on the Internet and get the one to meet users' needs. Thus, they have been useful tools for e-commerce companies (such as Amazon, e-bay, Google, Netflix, etc.) to provide automated and personalized suggestions of products to customers.

The recommender systems can be seen as information process systems due to the variated quantity of information processed to afford suggestions of products to users in an automated and personalized manner. The information processed by recommender systems may be explicit, (users' ratings), or implicitly, (users' behavior; applications downloaded; viewed or purchased items) [5]. Thus, data is primary about users and items. However, Ricci [1] refers to the data used by recommender systems as three kinds of objects:

- *Items*. The recommended objects.
- *Users*. The users of the system.
- *Transaction*. A recorded interaction between a user and the RS (ratings, reviews, etc.).

Letters U and I are frequently used to represent the set of users and items in the system, respectively. The set of possible values for a rating is represented by S and R is the set of rates recorded in the system. The interactions of users and items are commonly represented in a user-item rating matrix, see Fig. 1. Grey cells represent items rated by users and blank cells are not rated items. Typically, the number of users and items in the dataset are denoted by n and m respectively. Thus, it is an $n \times m$ matrix.

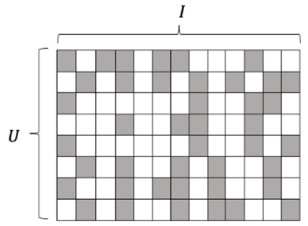

Figure 1
The user-item matrix. Grey cells represent users' ratings R

Even when there are many recommender systems approaches, collaborative filtering is widely used in online stores due to its proven success in this field [1, 2, 6, 12]. To recommend products to an active user, this approach considers the opinion of similar users to the active user about these products. Discovering those similar users is a challenging part of these methods. Thus, the selection of the appropriate similarity measure plays an important role in generating good recommendations, which is reflected in the active user's satisfaction.

Neighborhood-Based is a type of collaborative filtering method in which ratings gathered in the system are used right away to predict ratings for new items. There are two flavors for making the predictions: *User-Based* estimates the rating the user $u \in U$ would give to an item $i \in I$ by using the i's ratings given by other users v, best known as neighbors, which have a similar rating taste to user u. *Item-Based* estimate the rating the user u would give to an item i considering the rating user u

give to items $j \in I$ similar to i. In this case, two items are said to be similar if a considerable number of users have rated these items in a similar manner.

Fig. 2 illustrates the idea of Neighborhood-based collaborative filtering approaches; each row is a user's rating vector, and the columns are the items in the dataset. Grey cells represent items rated by users and blank cells are not rated items. Black cells represent the ratings used to predict the rating user u would give to an item i, represented by the striped cell.

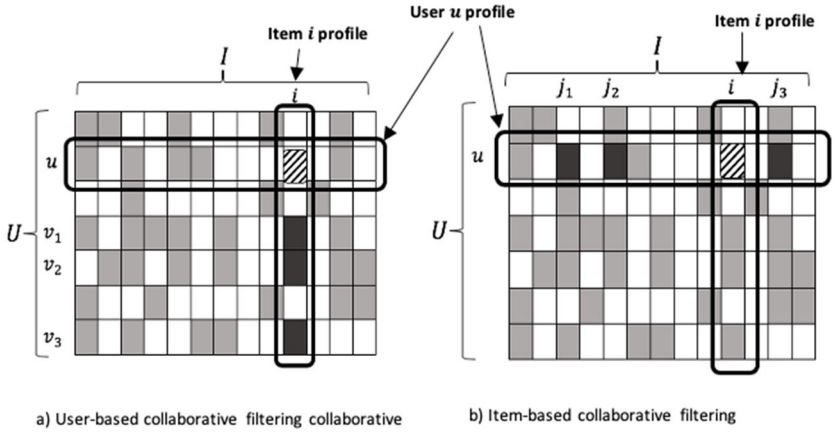

Figure 2
User-based and Item-based collaborative filtering to estimate the rating user u would give to an item i

2 The Cold-Start Problem

Collaborative filtering requires information about the user preferences to recommend or not an item. However, it is very common not to have enough information about users or items to create the recommendation, typically when they are new in the system. This problem is named the cold-start problem. [1, 12]

3 Similarity Measures in Recommender Systems

Similarity measures tell us how similar two objects are and quantify that similarity. Nevertheless, the similarity can also be given by their correlation [1]. The *k*-NN classifier is the preferred approach to collaborative filtering. This classifier is highly dependent on defining an appropriate similarity or distance measure. Hence, the choice of the appropriate similarity measure is the most critical component in these methods to make good recommendations.

Even when there are many similarity functions, the preferred ones in recommender systems are *Cosine* similarity and *Pearson's Correlation Coefficient* [1, 2]. However, they have been proven low performance in the recommendation problem domain. Therefore, new similarity measures have been proposed by many researchers to tackle the recommendation problem.

Recalling that users u and v are considered vectors of ratings. Let's now define r_{ui} and r_{vi} as the user-item rating given by user $u, v \in U$ to item $i \in I$. Then, I_u and I_v are the set of items rated by user u and v, respectively, and I_{uv} is the set of common rated items by both users. Hence, their *cosine* similarity can be expressed as the cosine angle that they form.

$$sim(u,v)^{COS} = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u} r^2{}_{ui}} \cdot \sqrt{\sum_{i \in I_v} r^2{}_{vi}}} \tag{1}$$

A drawback of cosine similarity is that it does not consider the differences in the mean and variance of the vectors u and v.

Pearson's Correlation Coefficient measures the linear relationship between the two vectors, users u and v rating vectors in recommender systems. It considers the average rating value of the two vectors u and v defined as $\bar{r}_u$ and $\bar{r}_v$ respectively.

$$sim(u,v)^{PCC} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) \cdot (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{2}$$

A modified version of equation (2) is the *Constrained Pearson's Correlation Coefficient* which was created to emphasize the effect of positive and negative ratings [4] by using the median of the rating scale. For instance, $r_{med} = 3$ on a scale from 1 to 5.

$$sim(u,v)^{CPCC} = \frac{\sum_{i \in I_{uv}} (r_{ui} - r_{med}) \cdot (r_{vi} - r_{med})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - r_{med})^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{vi} - r_{med})^2}} \tag{3}$$

The *Weighted Pearson's Correlation Coefficient* is another modified version of equation (2) which considers the common items between user u and v [13].

$$sim(u,v)^{WPCC} = \begin{cases} sim(u,v)^{PCC} \cdot \frac{|I_{uv}|}{H}, |I_{uv}| \leq H \\ sim(u,v)^{PCC}, otherwise \end{cases} \tag{4}$$

where H is an experimental value, it is set to 50 based on [2].

Another function that also considers the common items between user u and v is *Sigmoid Pearson's Correlation Coefficient* [16].

$$sim(u,v)^{SPCC} = sim(u,v)^{PCC} \cdot \frac{1}{1 + exp\left(-\frac{|I_{uv}|}{2}\right)} \tag{5}$$

It is very common that users tend to give low rates to items they like very much. Thus, the *Adjusted Cosine* measure was presented to consider the preference of the user's rating [14].

$$sim(u,v)^{ACOS} = \frac{\sum_{i\in I}(r_{ui}-\bar{r}_u)\cdot(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{ui}-\bar{r}_u)^2}\cdot\sqrt{\sum_{i\in I}(r_{vi}-\bar{r}_v)^2}} \quad (6)$$

Jaccard is another widely used measure. The main idea is that two users are more similar if they have more common ratings. However, it does not consider absolute ratings value.

$$sim(u,v)^{Jaccard} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (7)$$

So far, the state-of-art similarity measures were presented. However, new metrics have been proposed to address the recommender system problem. Shardanand proposed the *Mean Squared Difference* based on the mean squared difference distance [4, 8].

$$distance(u,v)^{MSD} = \frac{\sum_{i\in I_{uv}}(r_{ui}-r_{vi})^2}{|I_{uv}|} \quad (8)$$

Once the MSD distance is calculated, then all users whose $distance(u,v)^{MSD} < L$ are selected, and finally, the similarity is calculated using the following equation.

$$sim(u,v)^{MSD} = \frac{L-distance(u,v)^{MSD}}{L} \quad (9)$$

The problem with MSD is that it only considers the absolute rating, but it does not consider the percentage of common ratings. Nevertheless, Jaccard and MSD can be merged and form a metric that considers both absolute ratings and the percentage of common ratings [12], named *Jaccard Mean Squared Difference*.

$$sim(u,v)^{JMSD} = sim(u,v)^{Jaccard} \cdot sim(u,v)^{MSD} \quad (10)$$

A similarity measure proposed to alleviate the cold-statr problem in recommender systems is the one proposed by Ahn called *PIP* [6]. This measure is made-up of the following three factors of similarity:

1. *Proximity*, given two ratings, calculates the absolute difference between them and considers whether they are in agreement or not, giving a penalization to ratings in disagreement.
2. *Impact*, represents how strongly an item is accepted or refused by users.
3. *Popularity*, tells how common two users' ratings have. Two ratings can provide more information about the similarity of two users if the average rating of both users has an important difference from the average of total users' ratings.

Then, the PIP similarity between user u and v can be calculated using equation (11):

$$sim(u,v)^{PIP} = \sum_{i\in I_{uv}} PIP(r_{ui}, r_{vi}) \quad (11)$$

where $PIP(r_{ui}, r_{vi})$ is the PIP value for the two ratings r_{ui} and r_{vi} on item i by user u and v respectively. PIP can be defined by equation (12):

$$PIP(r_{ui}, r_{vi}) = Proximity(r_{ui}, r_{vi}) \cdot Impact(r_{ui}, r_{vi}) \cdot Popularty(r_{ui}, r_{vi}) \quad (12)$$

Liu proposed the *New Heuristic Similarity Model* (NHSM) which considers the common ratings, context information and it is normalized [2]. Liu improved PIP by taking advantage of the sigmoid function, which is a non-linear function, and it can penalize bad similarity or reward good similarity. The resulting function is named *Proximity Significance Singularity* (PSS):

- *Proximity*, considers the remoteness between two ratings.
- *Significance*, ratings are more significant when two ratings are further away from the median rating.
- *Singularity*, represents how two ratings are different with regard to other ratings.

The similarity measure of PSS is given by equations (13) and (14).

$$sim(u, v)^{PSS} = \sum_{i \in I_{uv}} PSS(r_{ui}, r_{vi}) \quad (13)$$

$$PSS(r_{ui}, r_{vi}) = Proximity(r_{ui}, r_{vi}) \cdot Significance(r_{ui}, r_{vi}) \cdot Singularity(r_{ui}, r_{vi}) \quad (14)$$

In addition, Liu also made use of a modified version Jaccard formula to penalize the small proportion of common ratings. The resulting modified Jaccard is given by equation (15):

$$sim(u, v)^{Jaccard'} = \frac{|I_u \cap I_v|}{|I_u| \times |I_v|} \quad (15)$$

Then equations (14) and (15) are combined as follows:

$$sim(u, v)^{JPSS} = sim(u, v)^{Jaccard'} \cdot sim(u, v)^{PSS} \quad (16)$$

Liu also considers each user's preference using equation (17). The idea behind is that some users might unwillingly give high scores to items they like, or vice versa.

$$sim(u, v)^{URP} = 1 - \frac{1}{1 + exp(-|\mu_u - \mu_v| \cdot |\sigma_u - \sigma_v|)} \quad (17)$$

where μ_u and μ_v represent the mean rating of user u and v respectively. The σ_u and σ_v are the standard variance of user u and v.

Finally, NHSM is the combination of equations (16) and (17):

$$sim(u, v)^{NHSM} = sim(u, v)^{JPSS} \cdot sim(u, v)^{URP} \quad (18)$$

4 Proposed Similarity Measures

The proposed similarity is inspired by the *Constrained Pearson's Correlation Coefficient* (CPCC), which emphasizes the positive and negative rates of the two users to calculate their correlation. As observed in equation (3), CPCC uses the center value of the scale, r_{med}, to distinguish when a rating is positive or negative.

Nevertheless, it has two disadvantages: it does not consider the proportion of common rated items of two users, and it may return low similarity even when there are many common rate values. For example, consider the data in Fig. 3 a). When calculating the users' CPCC similarity, presented in Fig. 3 b), the similarities seem not to be correct. As an example, $u1$ should have a high similarity with $u3$, instead, it has a high similarity with $u2$.

User	$i1$	$i2$	$i3$	$i4$
$u1$	4	3	5	4
$u2$	5	3		
$u3$	4	3	3	4
$u4$	2	1		
$u5$	4	2		

a) Users-Items-Ratings matrix taken from[2]

User	$u2$	$u3$	$u4$	$u5$
$u1$	1	0.577	-0.447	0.707
$u2$		1	-0.447	0.707
$u3$			-0.447	0.707
$u4$				0.316

b) CPCC

User	$u2$	$u3$	$u4$	$u5$
$u1$	0.5	0.577	-0.223	0.353
$u2$		0.5	-0.447	0.707
$u3$			-0.223	0.353
$u4$				0.316

c) JCPCC

User	$u2$	$u3$	$u4$	$u5$
$u1$	0.333	0.384	0	0.235
$u2$		0.5	0	0.707
$u3$			0	0.353
$u4$				0

d) PJCPCC

Figure 3
Users' similarities matrix for CPCC, JCPCC and PJCPCC

4.1 Jaccard Constrained Pearson's Correlation Coefficient

CPCC can be multiplied by *Jaccard* similarity measure, equation (7), to give it support. The resulting similarity measure is named *Jaccard Constrained Pearson's Correlation Coefficient*, JCPCC.

$$sim(u,v)^{JCPCC} = sim(u,v)^{Jaccard} \cdot sim(u,v)^{CPCC} \tag{19}$$

4.2 Positive Constrained Pearson's Correlation Coefficient

Continuing with the example, Fig. 3 c) displays the similarities using JCPCC. Now, $u1$ has the highest similarity with $u3$.

However, suggesting items to the active user u for which it has a positive response is the aim of recommender systems. Therefore, the Jaccard similarity can only consider the common rates whose value is positive. A rating r_{ui} is positive if $r_{ui} \geq \theta$, for instance, $\theta = 4$ on a scale from 1 to 5 [17, 18, 19]. This similarity is named *Positive Jaccard Constrained Pearson's Correlation Coefficient*, PJCPCC. In equation (20), the parameter θ is used to filter the positive rates in u and v.

$$sim(u,v)^{PJCPCC} = sim(u,v,\theta = 4)^{Jaccard} \cdot sim(u,v)^{CPCC} \quad (20)$$

The resulting similarities using PJCPCC are then displayed in Fig. 3 d). The similarities now are adjusted based on the common items rated positively and the correlation given by their rates.

5 Datasets

For the experiments, two of the most used datasets from Movie Lens were selected (https://grouplens.org/): ML-100K and ML-Latest-Small. Table 1 illustrates the datasets information and Fig. 4 illustrates the rating distribution of the datasets.

Table 1
Datasets information

Dataset	Ratings	Users	Items	Density $\frac{\#\text{Rates} * 100}{\#\text{Users} * \#\text{Items}}$	Rate Scale
ML-100K	100,000	944	1,682	6.3%	[1, 2, 3, 4, 5]
ML-Latest-Small	100,836	610	9724	1.7%	[0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]

The test dataset is created by following the next steps:

1. The users' set U is randomly split: 80% for training (U_{train}), and 20% for testing (U_{test}), the users to which the system creates recommendations.
2. To simulate a cold-start behavior, only ten ratings are randomly selected as training ratings (R_{train}) for each user in testing (U_{test}). The remaining rates are the testing ratings (R_{test}), rates to be predicted and evaluated.

The whole process is done using k-folds cross-validation with $k = 5$. Thus, each fold contains a disjointed user test set, U_{test}, with the training rates R_{train} of each user $u \in U_{test}$. The set of items rated in R_{train} by user u is denoted as I_{train}.

Similarly, the set of items rated in R_{test} by user u is denoted as I_{test}. Hence, the set of all items rated by user u is denoted as $I_u = I_{train} + I_{test}$.

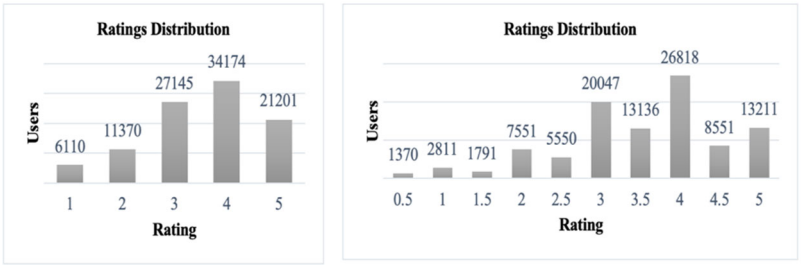

Figure 4
a) ML-100K and b) ML-Latest-Small datasets ratings distribution

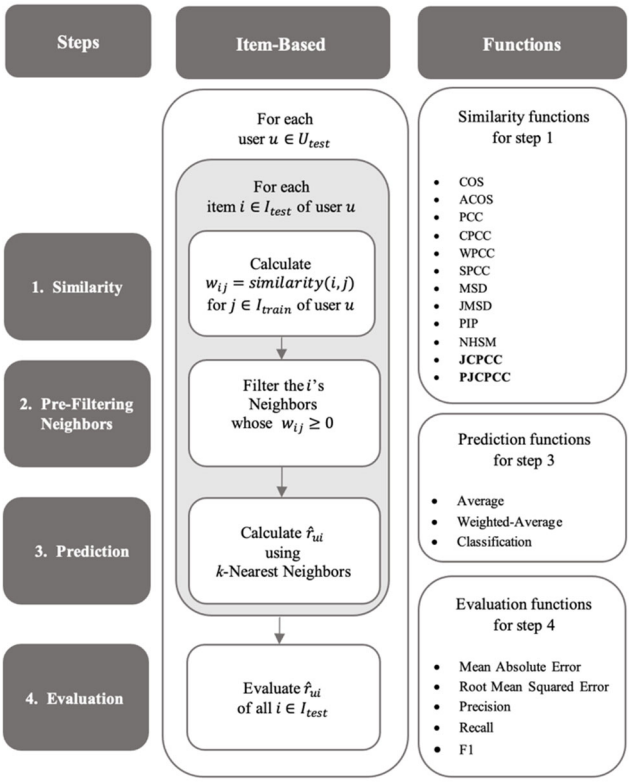

Figure 5
Collaborative filtering Prediction-Evaluation framework

6 Collaborative Filtering Prediction-Evaluation Framework

The collaborative filtering prediction-evaluation framework has four steps in general: similarity, pre-filter neighbors, prediction, and evaluation. The collaborative filtering prediction-evaluation framework used in this research is illustrated in Fig. 5. The Functions column displays the evaluated similarity functions for step 1, the prediction for step 3, and the evaluation functions for step 4. The neighbors are pre-filtered by their similarity weight $w_{ij} \geq 0$ for step 2.

7 Results

The results are dived into two sections for ML-100K and ML-Latest-Small datasets. Each section contains the graphs to illustrate the similarity measures performance using *MAE*, *RMSE*, *Precision*, *Recall*, and *F1* evaluation metrics for each rating prediction function: *Average*, *Weighted-Average*, and *Classification*. The rating predictions were calculated using the *k*-Nearest-Neighbors' ratings, for k = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] because users in testing have up to ten ratings for training.

7.1 ML-100K Results

7.1.1 Average Rating Prediction

Fig. 6 illustrates the MAE, RMSE, precision, and recall results for *Average* rating prediction. The similarity measures with the lowest MAE are the proposed JCPCC and PJCPCC. Their MAE value is close to 0.94 when $k = 1$ and it decreases as k increases until they reach the lowest error, close to 0.84 when $k = 6$ and $k = 7$. For RMSE, JCPCC and PJCPCC are in the group of similarity measures with the lowest RMSE value, around 1.26 when $k = 1$. This value also decreases as k increases, which is about 1.05 when $k = 6$. Regarding precision, JCPCC has the highest precision value, which is about 0.64 when $k = 1$, and it is followed by PJCPCC whose value is close to 0.63. In general, the precision decreases as long as k increases, and these two similarities are affected. Regarding recall, JCPCC and PJCPCC have the highest values, which is about 0.58 when $k = 1$.

In general, recall decreases as long as k increases, but JCPCC and PJCPCC keep the highest value. Finally, Fig. 7 illustrates the F1 metric. It can be observed that the proposed JCPCC and PJCPCC have the highest value, about 0.6 when $k = 1$. This value also decreases as long as k increases. However, JCPCC and PJCPCC are in the group of similarity measures with the highest F1 value.

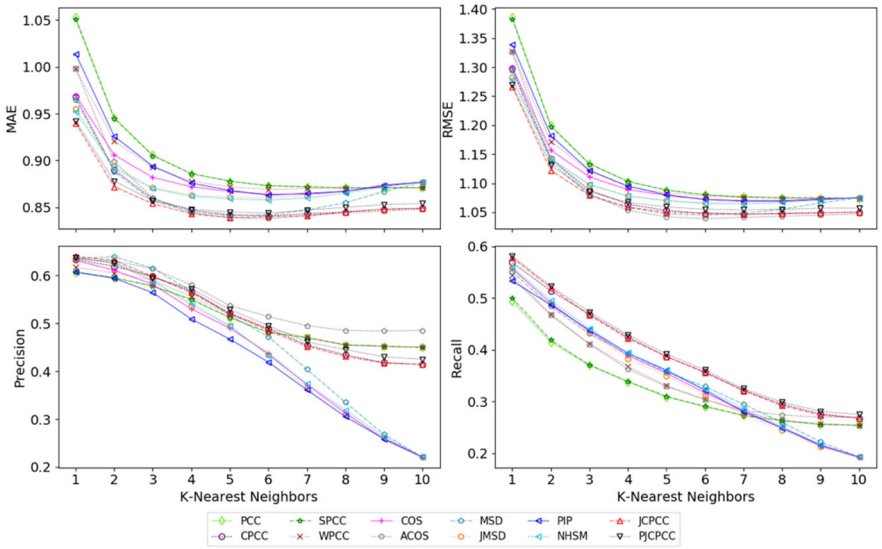

Figure 6
MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Average rating prediction on ML-100K dataset

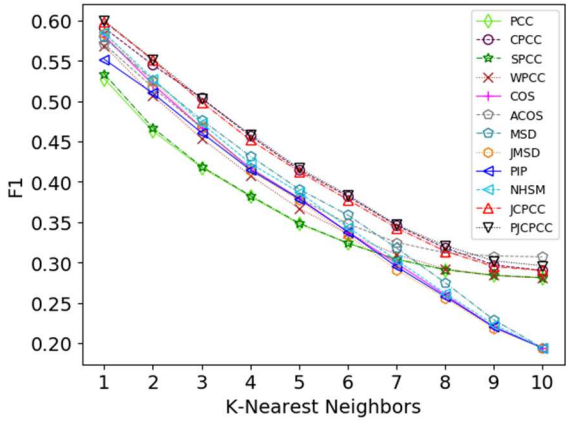

Figure 7
F1 vs K-Nearest Neighbors using Average rating prediction on ML-100K dataset

7.1.2 Weighted Average Rating Prediction

Fig. 8 illustrates the MAE, RMSE, precision, and recall results for *Weighted-Average* rating prediction. The similarity measures with the lowest MAE is the proposed JCPCC and PJCPCC similarities. Their MAE value is close to 0.94 when $k = 1$ and it decreases until they reach the lowest error, close to 0.82 when $k \geq 7$.

With respect to RMSE, JCPCC has the lowest error, around 1.26 when $k = 1$. This value decreases below 1.05 when $k \geq 5$. Regarding precision, the two similarities are in the group of the highest values, about 0.64 when $k = 1$. As observed, the precision decreases as long as k increases. However, PJCPCC keeps the highest precision value. With regard to recall, JCPCC and PJCPCC have also the highest values, which is about 0.58 when $k = 1$. Even when recall decreases as long as k increases, PJCPCC keeps the highest value.

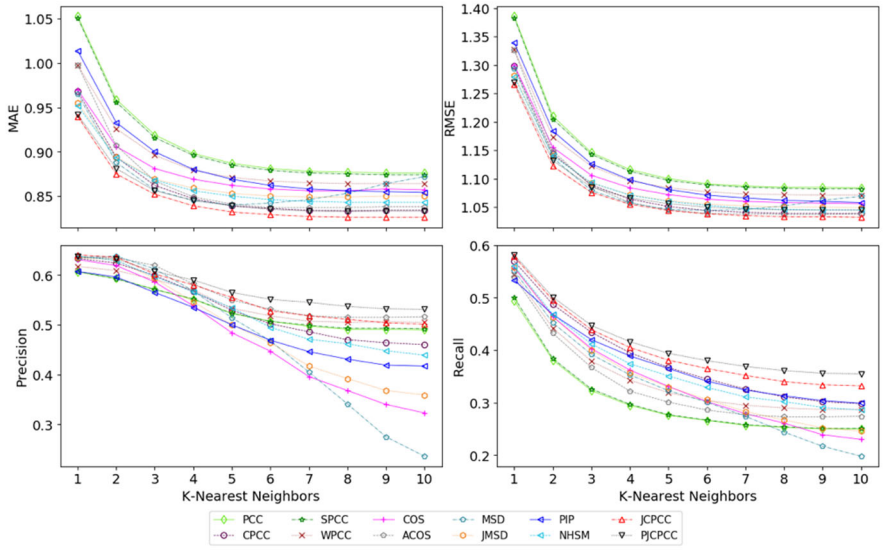

Figure 8

MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Weighted Average rating prediction on ML-100K dataset

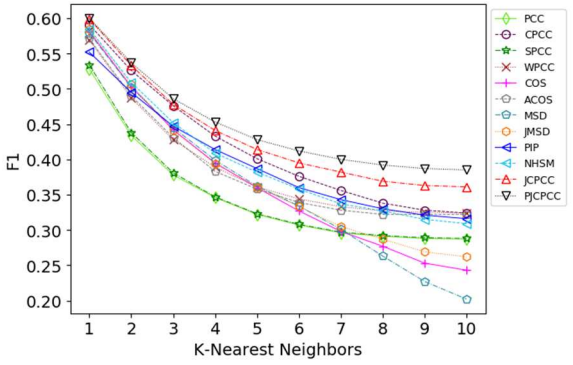

Figure 9

F1 vs K-Nearest Neighbors using Weighted Average rating prediction on ML-100K dataset

Finally, Fig. 9 illustrates the F1 metric. It can be observed that the proposed JCPCC and PJCPCC have the best performance for all k values. F1 value is about 0.6 when $k = 1$ and it decreases as long as k increases. However, JCPCC always has the highest value.

7.1.3 Classification Rating Prediction

Fig. 10 illustrates the MAE, RMSE, precision, and recall results for *Classification* rating prediction. As observed, both similarity measures, PJCPCC and JCPCC, have the best performance. The lowest MAE and RMSE errors, and the highest precision and recall values.

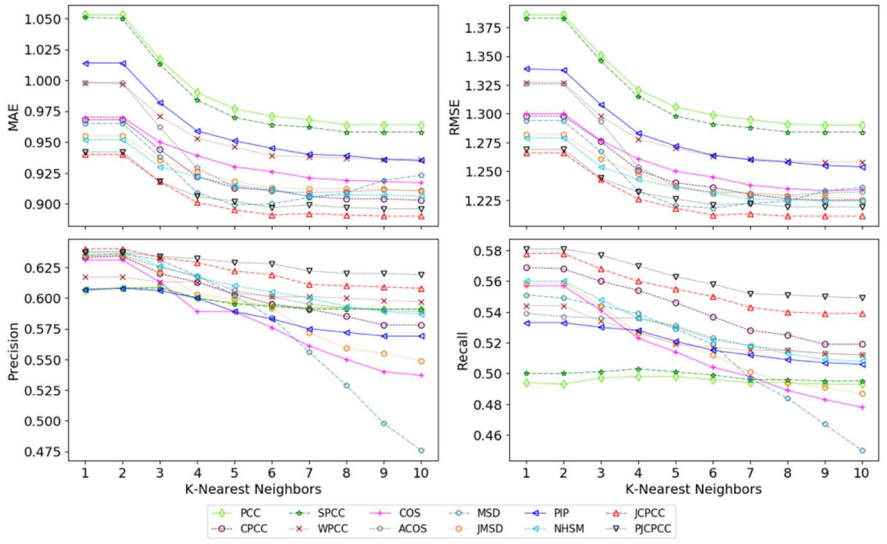

Figure 10
MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Classification rating prediction on ML-100K dataset

MAE value is close to 0.94 when $k = 1$, and $k = 2$, then it decreases until it reaches the lowest error close to 0.89 when $k \geq 6$. Similar behavior is observed for RMSE, whose value is about 1.26 when $k = 1$ and $k = 2$. Then, it decreases until it reaches the lowest error close to 1.211 when $k \geq 8$. With regard to precision, PJCPCC and JCPCC have the highest value, about 0.625, which reminds stable regardless of the k value. Regarding recall, PJCPCC and JCPCC also have the best performance. However, PJCPCC has the highest value, around 0.58 when $k \leq 2$ and it decreases to about 0.55 when $k \geq 7$.

Finally, Fig. 11 illustrates the F1 metric. In this case, JCPCC and PJCPCC have the best performance whose highest value is about 0.6 when $k \leq 2$. Even when F1 decreases while k increases, PJCPCC holds the highest F1 values and it is followed by JCPCC with the second-highest F1 value.

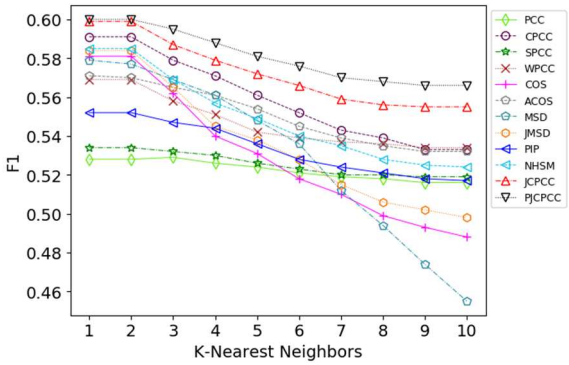

Figure 11
F1 vs K-Nearest Neighbors using Classification rating prediction on ML-100K dataset

7.2 ML-Latest-Small Results

7.2.1 Average Rating Prediction

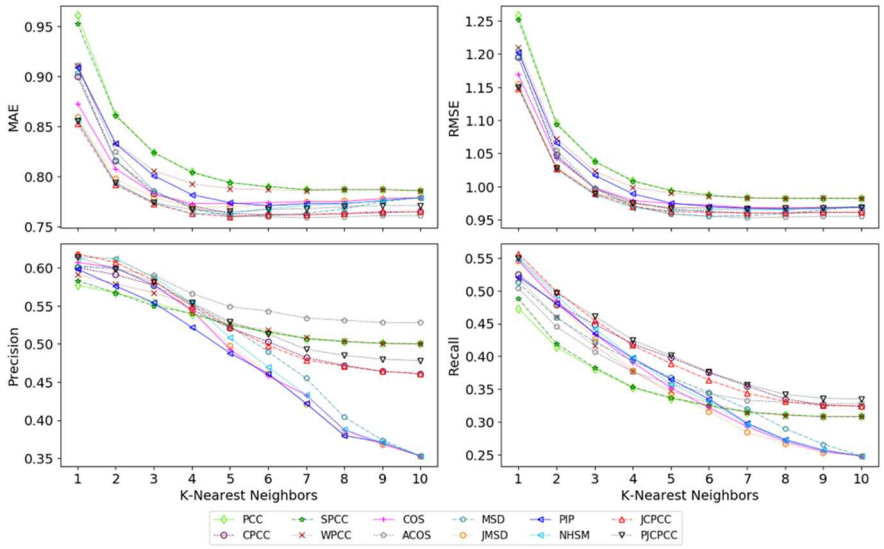

Figure 12
MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Average rating prediction on ML-Latest-Small dataset

Fig. 12 illustrates the MAE, RMSE, precision, and recall results for *Average* rating prediction. In general, JCPCC and PJCPCC are in the group of similarities with the lowest MAE. Their MAE value is close to 0.85 when $k = 1$ and it decreases until

they reach the lowest error, which is close to 0.76 when $k \geq 4$. For RMSE, JCPCC and PJCPCC also have the lowest error, around 1.15 when $k = 1$. They are also in the group of similarities with the lowest RMSE as k increases. Regarding precision, JCPCC has the highest precision value, about 0.61 when $k \leq 2$. In general, the precision decreases as long as k increases. With regard to recall, JCPCC and PJCPCC have the highest values, which is about 0.55 when $k \leq 2$, even when recall decreases while k increases, both similarity measures hold the highest recall values. Finally, Fig. 13 illustrates the F1 metric. The similarities JCPCC and PJCPCC have an F1 close to 0.57 when $k = 1$. When k increases F1 decreases, but the two similarities are in the group of similarity measures with the highest F1 values.

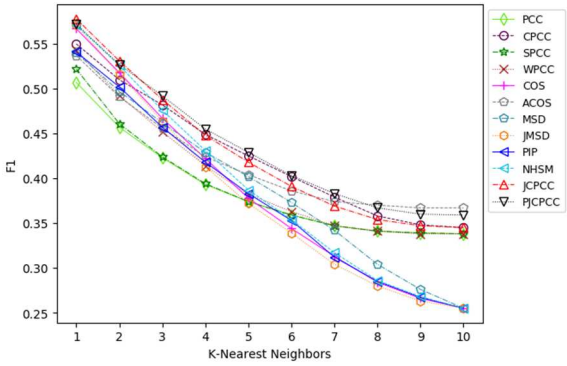

Figure 13
F1 vs K-Nearest Neighbors using Average rating prediction on ML-Latest-Small dataset

7.2.2 Weighted Average Rating Prediction

Fig. 14 illustrates the MAE, RMSE, precision, and recall results for *Weighted-Average* rating prediction. JCPCC, PJCPCC are in the group of similarity measures with the lowest MAE. Their MAE value is close to 0.85 when $k = 1$ and it decreases until they reach the lowest error, close to 0.75 when $k \geq 6$. Similarly, JCPCC and PJCPCC are also in the group of similarity measures with the lowest RMSE error, around 1.15 when $k = 1$. This value also decreases as k increases, but JCPCC and PJCPCC hold a good performance, about 0.95 when $k \geq 6$. Regarding precision, JCPCC and PJCPCC are in the group of similarity measures with the highest precision value, about 0.61 when $k \leq 3$. In general, the precision decreases as long as k increases. Similarly, JCPCC and PJCPCC have also in the group of similarity measures with the highest recall, which is about 0.55 when $k = 1$. Even when recall decreases as long as k increases PJCPCC reminds as the similarity with the highest value.

Finally, Fig. 15 illustrates the F1 metric. JCPCC has the highest value, about 0.58 when $k = 1$. It is followed by PJCPCC with a value close to 0.57. As long as k increases, the F1 value decreases for all similarity measures, but PJCPCC keeps the highest value.

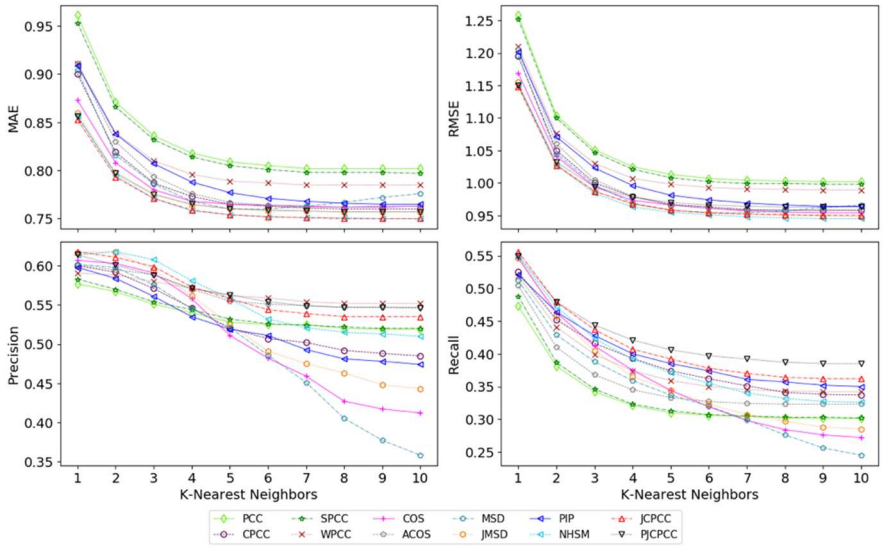

Figure 14

MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Weighted Average rating prediction on ML-Latest-Small dataset

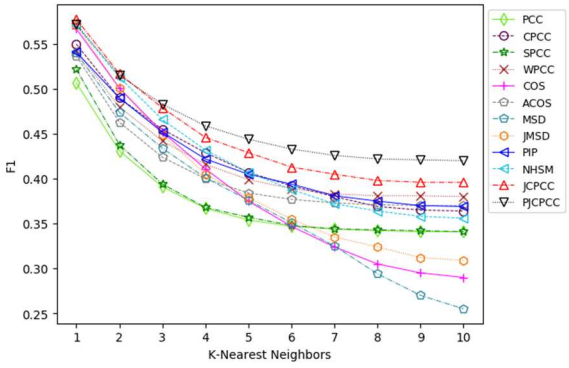

Figure 15

F1 vs K-Nearest Neighbors using Weighted Average rating prediction on ML-Latest-Small dataset

7.2.3 Classification Rating Prediction

Fig. 16 illustrates the MAE, RMSE, precision, and recall results for *Classification* rating prediction. The results are similar to those for this scenario when using the ML-100K dataset. In general, both similarity measures, PJCPCC and JCPCC, have the best performance. The lowest MAE and RMSE errors, and the highest precision and recall values. MAE value is close to 0.86 when $k = 1$ and $k = 2$, then it decreases until it reaches the lowest error close to 0.81 when $k \geq 7$.

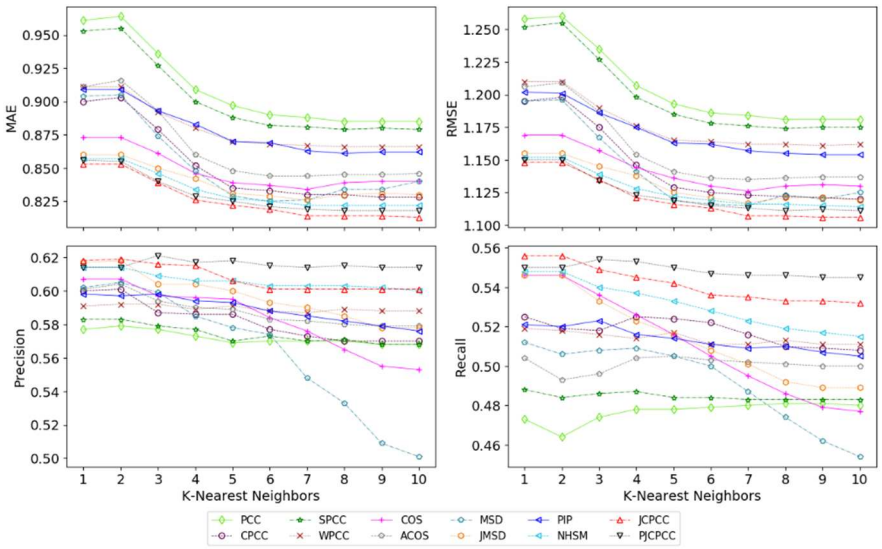

Figure 16

MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Classification rating prediction on ML-Latest-Small dataset

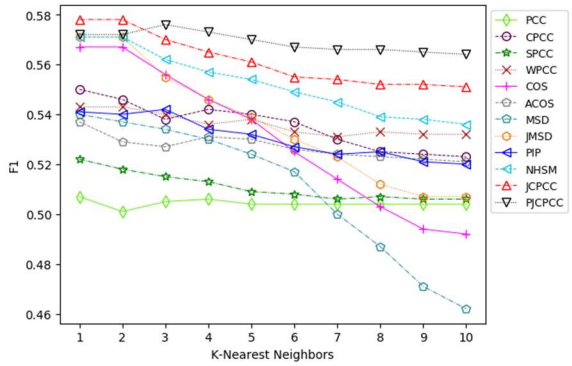

Figure 17

F1 vs K-Nearest Neighbors using Classification rating prediction on ML-Latest-Small dataset

Similar behavior is observed for RMSE, whose value is about 1.15 when $k = 1$ and $k = 2$, and then it decreases until it reaches the lowest error close to 1.1 when $k \geq 7$. It is also observable that PJCPCC and JCPCC precision value reminds stable regardless of the k value, which is about 0.61. Regarding recall, PJCPCC and JCPCC have the best performance. However, JCPCC has the highest value, around 0.55 when $k \leq 2$, and then PJCPCC is the one with the highest value about 0.54 $k \geq 3$ and this value reminds stable regardless of the k value.

Finally, Fig. 17 illustrates the F1 metric. In this case, JCPCC and PJCPCC have again the best performance. When $k \leq 2$ JCPCC has the highest F1 value, about 0.58. In this k range, PJCPCC's F1 value is about 0.57. It can be observed that when $k \geq 3$, PJCPCC keeps the highest value, which is about 0.57. In this k range, JCPCC now has the second-highest F1 value, which is between 0.55 and 0.54.

Conclusions

This paper analyses the similarity measures used to address the cold-start problem in Recommender Systems; when there is a small amount of information about users' preferences. In addition, it proposes two similarity measures to address this problem, JCPCC, and PJCPCC. The new similarity measures were analyzed and compared with state-of-art and other similarity measures using the *Item-Based* collaborative filtering approach, and different rating prediction functions (*Average*, *Weighted-Average,* and *Classification* rating predictions). The experiments were performed using two different MovieLens datasets, and the evaluation metrics used were MAE, RMSE, precision, recall, and F1. The results produced by the experiments proved that these new similarity measures could yield better performance than other measures used in cold-start scenarios in *Item-Based* collaborative filtering approaches.

Acknowledgment

The work is partially supported by projects 20220857 and 20211874 of SIP IPN, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Deep Learning Platform for Language Technologies of the Supercomputing Laboratory of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America Ph.D. Award.

References

[1] F. Ricci, L. Rokach, B. Shapira, and K. P. B., Recommender Systems Handbook, First. Boston, MA: Springer US, 2011

[2] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," Knowledge-Based Syst., Vol. 56, pp. 156-166, 2014

[3] Y. Wang, J. Deng, J. Gao, and P. Zhang, "A hybrid user similarity model for collaborative filtering," Inf. Sci. (Ny), Vol. 418-419, pp. 102-118, 2017

[4] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating 'word of mouth,'" Conf. Hum. Factors Comput. Syst. - Proc., Vol. 1, pp. 210-217, 1995

[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Syst., Vol. 46, pp. 109-132, 2013

[6] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," Inf. Sci. (Ny), Vol. 178, No. 1, pp. 37-51, 2008

[7] J. Bobadilla, A. Hernando, F. Ortega, and J. Bernal, "A framework for collaborative filtering recommender systems," Expert Syst. Appl., Vol. 38, No. 12, pp. 14609-14623, 2011

[8] P. Maes and F. R. Morgenthaler, "Social Information Filtering for Music Recommendation," Master Thesis, 1994

[9] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-N recommendation tasks," RecSys'10 - Proc. 4th ACM Conf. Recomm. Syst., No. January 2010, pp. 39-46, 2010

[10] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data," Knowledge-Based Syst., Vol. 82, pp. 163-177, 2015

[11] I. Batyrshin, "Towards a general theory of similarity and association measures: Similarity, dissimilarity and correlation functions," J. Intell. Fuzzy Syst., Vol. 36, No. 4, pp. 2977-3004, 2019

[12] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," Knowledge-Based Syst., Vol. 26, pp. 225-238, 2012

[13] J. L. Herlocker, Joseph A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," ABA J., Vol. 102, No. 4, pp. 24-25, 2017

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," Proc. 10th Int. Conf. World Wide Web, WWW 2001, pp. 285-295, 2001

[15] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," Found. Trends Human-Computer Interact., Vol. 4, No. 2, pp. 81-173, 2010

[16] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in Proceeding of the ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, 2009, pp. 397-405

[17] I. Z. Batyrshin, "Constructing correlation coefficients from similarity and dissimilarity functions," Acta Polytechnica Hungarica, Vol. 16, No. 10, pp. 191-204, 2019

[18] I. Batyrshin, F. Monroy-Tenorio, A. Gelbukh, L. A. Villa-Vargas, V. Solovyev, and N. Kubysheva, "Bipolar rating scales: A survey and novel correlation measures based on nonlinear bipolar scoring functions," Acta Polytechnica Hungarica, Vol. 14, No. 3, pp. 33-57, 2017

[19] F. Monroy-Tenorio, I. Batyrshin, A. Gelbukh, V. Solovyev, N. Kubysheva, and I. Rudas, "Correlation measures for bipolar rating profiles," in Fuzzy Logic in Intelligent System Design, Springer, pp. 22-32, 2018

A Joint Algorithm for Base Station Deactivation and Mobile User Reassignment in Green Cellular Networks

Anabel Martínez-Vargas[1], Ángel G. Andrade[2], Zury J. Santiago-Manzano[1]

[1] Universidad Politécnica de Pachuca, Carretera Pachuca - Cd. Sahagún km 20 Ex-Hacienda de Santa Bárbara, 43830, Zempoala, México, anabel.martinez@upp.edu.mx, zury_santiago@micorreo.upp.edu.mx

[2] Facultad de Ingeniería; Universidad Autónoma de Baja California, Boulevar Benito Juárez S/N, 21280, Mexicali, México, aandrade@uabc.edu.mx

Abstract: The proliferation of smartphones has led to an increase in the cellular infrastructure, due to efforts by mobile operators to meet the rising demand. Given that the planning of cellular networks is carried out according to demand during peak hours, a large number of base stations must be deployed to maintain a constant number of base stations even when traffic intensity is reduced. This strategy has brought about increased energy levels in cellular networks, affecting the networks' operating expenses and contributing to the problem of carbon emissions in the atmosphere. This work shows an algorithm that deactivates base stations for cellular networks and reassigns mobile users. We use the interruption probability to analyze the effect of base-station-deactivation on mobile users. We perform two approaches: one using a homogeneous network and the other a heterogeneous network. The homogeneous network is a macro-cell deployment, whereas the heterogeneous network comprises macro-cells and femto-cells. A genetic algorithm is used to find the set of base stations to deactivate and continue offering the demand services. As the carrier-to-interference ratio increases, the results show that few base stations need deactivating in a heterogeneous network with high traffic demand.

Keywords: Genetic algorithm; Green network; Sleep mode

1 Introduction

Currently, there are mobile applications for almost any activity performed on smartphones, from carrying out banking operations to measuring kilometers traveled during a walk. The applications work automatically, anywhere, and at any time. As a result, smartphones have gained popularity among the world's population, as shown by the fact that, in 2018, 66% of the said population had a

mobile device. Following this trend, it is estimated that by the year 2023, the percentage will increase to 71% [1]. These devices provide voice and text messaging services and focus on online services such as data storage and social networks, as well as music and video transmissions, all of which generate more data traffic.

Network operators' strategy to meet the demand for data is to increase the number of base stations in a network [2]. This proposal is known as network densification [3]. It implies that many base stations (BSs) are deployed to handle high traffic status. However, the same number of active BSs is maintained even when traffic intensity is reduced. BSs use between 60% and 80% of the total energy utilized in a cellular network [4], and are responsible for 70% of the network's carbon dioxide emissions, making BSs the most energy-consuming devices in a network [5]. On a global scale, the information and communications technology industry contribute 2% to the world's CO_2 emissions [6].

In essence, there is one important reason why the development of green cellular networks has been proposed to address the imbalance between energy performance and energy consumption: the need for environmentally friendly cellular networks [7]. Researchers in the communications industry have focused on improving energy efficiency because BSs are the primary consumers of energy in a cellular network. Some solutions to reduce energy consumption involve BS hardware modifications or intelligent management of the elements of a network based on variations in traffic load [7]. Other solutions, reported in [7], propose reducing power amplifier operation periods, deploying heterogeneous networks, or switching BSs on/off.

The number of active BSs can be optimized by shutting down underused BSs and loading all users from the off BSs to the active BSs through a reassignment process [8]. To deactivate BSs within a network, it is necessary to find the minimum set of active BSs needed to continue offering the services in demand. This problem is not a trivial problem, given that various factors influence which BS should be active, such as radius coverage of the BS, available channels, and interference. On the other hand, the significant number of BSs in a network increases the possible combinations of active BSs, i.e., possible solutions to the problem. Therefore, minimizing active BSs in a cellular network is considered an NP-Hard [9] type problem since the time spent looking for a feasible solution is substantial. It should be mentioned that the optimal solution to this problem has not been found because there are potentially many, and the solution depends on multiple factors.

User reassignment adds complexity to the problem. A decision must be made about which subset of BSs to disable and which mobile users to associate with each active BS. These are yet more factors to consider when modeling the system.

The protocol for the assignment of mobile users should not be confused with that of their reassignment. The former has already been widely studied and reflected in standards mentioned in [4] [10] [11], whose improvements are focused on energy

efficiency and load balancing. Whereas the latter, the user-reassignment, arises due to the BS being shut down.

The present work proposes an algorithm for BS deactivation and mobile user reassignment. The algorithm uses an optimization model designed to minimize the number of active BSs by using the fundamental processes of an artificial intelligence technique called a Genetic Algorithm (GA) [12]. The use of GAs has proven to be appropriate in the context of this research topic, as shown in [2] and [6]. Reassigning users is an extensive process as all BSs search for the best service conditions for their users. For this reason, we apply a steady-state population model [12] of the GA to reduce the number of times the processes of crossover, mutation, and selection are carried out.

The GA for deactivating BSs and reassigning mobile users is responsible for evaluating the network at a given moment. It finds the minimum set of active BSs needed and performs the user reassignment process to maintain a low interruption probability (PI) value. To achieve this, in the optimization model, we explicitly consider the PI to analyze how BS deactivation affects users' service. The energy saved in this type of algorithm will depend on the number of deactivated BSs. If many BSs are deactivated, the energy saved can be substantial [7].

This paper is organized as follows: Section 2 presents the related work. Section 3 describes the system and optimization models. Section 4 explains the base station deactivation and mobile user reassignment algorithm. Section 5 discusses our experiments and their results. Finally, we present the conclusion, and address implications for further research.

2 Related Work

In the existing literature, several papers seek to reduce energy consumption in cellular networks by minimizing the number of active BSs. For example, in [9], an optimization framework is proposed that chooses a minimum set of BSs. It allocates mobile users accordingly (reassignment process) while meeting their target Signal-to-Interference-plus-Noise Ratio (SINR) constraints. It involves two approaches: the proactive approach and the reactive approach. The former begins with a low traffic load. As traffic increases, BSs are turned on. The latter starts with a high traffic load, and BSs are turned off as traffic decreases. The problem is transformed to one of full linear programming for small and medium networks and is solved with a branch and bound algorithm. For larger networks, a heuristic solution is proposed: each time the algorithm tries to eliminate a BS, it constructs a new Voronoi tessellation, calculates the SINR for each BS and mobile user pair, and also calculates the interruption probability. Unlike [9], our algorithm can be adapted to any network size. It considers a PI threshold in the model.

In [2], BSs shut down in a specific order, not necessarily beginning with the lower load BSs, and a smaller number of BSs are allowed to remain active. The authors propose an approach to minimize the number of active BSs. The reassignment process assigns a mobile user to a BS with the highest spectral efficiency without violating the bandwidth constraints; otherwise, the mobile user is blocked. It is a centralized cell zooming approach based on work in [13], in which a GA finds an ordering which results in more BSs being switched off. Even though cell zooming techniques focus on the energy consumption of the whole network, they may cause inter-cell interference and gaps in coverage [13]. In contrast, our algorithm does not deactivate BSs because they have smaller loads. Instead, it leaves active those BSs that can receive more users from their neighboring BSs, i.e., those located in areas with more users.

Similarly, work in [14] applies a binary Social Spider algorithm to solve the problem of BS deactivation by minimizing the number of active BSs. To shut down the BSs, the algorithm penalizes the fitness function according to the cell traffic load of the available neighboring BSs. If the neighboring BSs can serve the traffic load initially handled by deactivated BSs, the penalty value is lower; if not, it is higher. They do not include PI constraint in the original optimization problem. Our algorithm also applies a penalty function, but, unlike the work in [14], we increase the fitness value of a candidate solution if it cannot achieve the PI threshold. We guarantee that the BS set selected by the GA serves 99% of mobile users.

Work in [4] couples its approach to BS deactivation with user association. It proposes a fitness function that minimizes the trade-off between energy consumption and flow-level performance. Two problems arise from this: 1) a user association problem for which a policy is defined that guarantees that mobile users associate with the BS in an energy-efficient manner, taking into account the load balance; 2) a BS switching on/off problem that is solved employing a greedy algorithm. On the other hand, our algorithm evaluates the network-wide impact of BS deactivation on mobile users using the PI.

An algorithm that switches BSs off and on in a heterogeneous network (cellular network and wireless local area network) has been proposed [10]. Its cost function minimizes energy consumption and maximizes network revenue. To make it tractable, the authors divide the problem into two sub-problems (user association and BS switching on/off). On the one hand, the user association algorithm connects users to BSs or access points (APs) depending on their energy efficiency and revenue. For the BS deactivation problem, work in [10] proposes two greedy algorithms: the first one is based on the cost function (it turns off the BS that yields the maximum cost gain), and the second one is based on the density of access points within the coverage of each BS (it turns off the BS with the most significant number of mobile users associated with APs). However, this approach does not evaluate the impact of the switching on/off strategy on mobile users or Quality of Service (QoS) degradation. Moreover, greedy algorithms have a very high computational cost [15].

Work in [6] proposes to resize an LTE green network, determining the minimum number of active BSs needed given a specific traffic load, with restrictions on QoS. The number of active BSs represents energy reduction. A random number representing the number of active BSs is generated and, based on already proposed disconnection patterns, the active BSs are selected. A GA is applied to solve the problem. The user association or reassignment process is considered in the optimization model. Their approach introduces user outage per BS in the optimization model instead of presenting it at the network system level.

The work presented in [11] uses a BS on/off algorithm to reduce energy consumption in a cellular network. It establishes that BSs be deactivated one at a time since this minimally affects the load of the other BSs. Each time a BS is turned off, the load increment in neighboring BSs is evaluated. To do this, the algorithm considers the type of region (urban, metropolitan, etc.), the location of the BS, and its coverage. It proposes a sequential algorithm called Switching-on/off based Energy Saving (SWES). It is based on sharing information (feedback) between BSs and mobile users, such as system load and signal strength. When a BS is switched off, users are reassigned to the new BS with the second-best signal strength. However, the feedback may generate a large amount of data to send along with the information required by each user. Additionally, it does not quantify how mobile users are affected by the switching-off process (user outage).

The studies mentioned above addresses the reassignment process either jointly with or separately from the BS deactivation algorithm. We propose a joint algorithm for base station deactivation and mobile user reassignment, but, as opposed to the works discussed above, we use the PI metric in the optimization model to quantify how the BS switching-off process affects the mobile users in a network system. Then, in the GA processes, we add a penalty function to increase the fitness value of a candidate solution if it cannot achieve the PI threshold imposed in the optimization model. We guarantee that the BS set selected by the GA serves 99% of mobile users. Also, unlike the previous works, we analyze the performance of our proposed approach in heterogeneous and homogeneous networks at different traffic loads (number of mobile users). Another difference is that our work exploits the spectrum sharing approach to reuse frequencies in BSs. This efficiently exploits the channels available in a given BS since a set of mobile users can transmit over the same channel simultaneously [16].

3 System Model

Fig. 1 shows a cellular system network composed of several BSs and mobile users deployed over a two-dimensional area. Each base station (BS_j) and mobile user (UT_i) have random Cartesian coordinates that follow a uniform distribution. To differentiate the coordinates of these two components, a BS_j uses the notation

(x_j, y_j); on the other hand, a UT_i uses (u_i, v_i). The total BSs and mobile users in a network at a given moment are indicated by J and I, respectively.

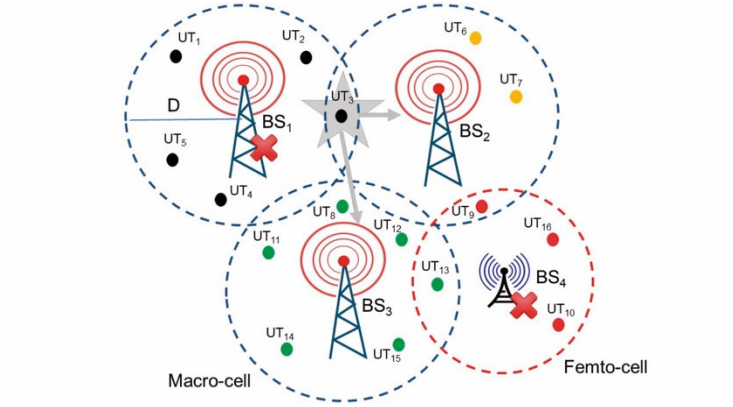

Figure 1
System scenario

Mobile users (UTs) assigned to a BS are delimited within their coverage radio D (see Fig. 1). The BS can be one of two types: macro-BS and femto-BS. The coverage radius of a femto-BS will always be less than the coverage radius of a macro-BS. When a BS is switched off (see BS_1 and BS_4 in Fig. 1) and some UTs linked to it cannot be reassigned to a new BS, the UTs are considered without service.

The Euclidean distance between a BS_j and a UT_i is denoted as $d_{i,j}$ and is calculated by applying Equation 1:

$$d_{i,j}(km) = \sqrt{\left(u_i - x_j\right)^2 - \left(v_i - y_j\right)^2} \tag{1}$$

Each BS_j provides service to several UT_i simultaneously; to know this relationship, the User-Base Station Relationship (RBU) matrix was created, as shown in Fig. 2, where the rows represent the BS_j and the columns represent the UT_i. If $RBU_{j,i} = 1$, the BS_j serves the user UT_i; otherwise, there is no relationship between BS_j and UT_i. In this way, a switched-off BS is made evident, as in the case of BS_4 (row 4), since there are only zeros in its elements.

It is also possible to know which UTs are not associated with any BS, as in the case of UT_{10}, since all the cells that represent it have values of zero. A BS can only allocate a specific number of C channels and service a certain number of UTs. MTU is the maximum number of UTs that a BS can serve. Macro-cells can serve more mobile users than femto-cells.

The binary vector Solutions for BS control (SBS) represents a GA's candidate solution (individual). Its length is equal to the value of J. The BS_j is switched-on if the SBS_j element has a value of 1 and turned off otherwise.

Base Station Number	Mobile Users														
	1	2	3	4	5	6	7	8	9	10	11	12	13	...	*I*
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2
RBU matrix

Fig. 3 shows an individual and the scenario it refers to; it proposes that BS_2, BS_3, BS_4 remain switched on. On the other hand, the vector CU of a length equal to I contains the channel identifier that each UT_i has been assigned to by the BS that serves it. The elements in CU can take a value from 1 to C. It is essential to mention that index k refers to an individual or SBS vector specific to the GA population. Index j refers to one particular BS and index i is a particular UT of the network.

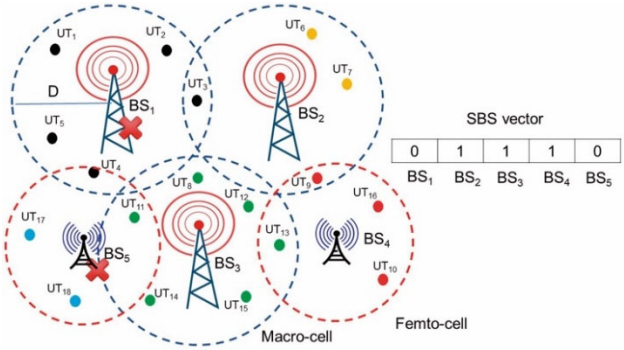

Figure 3
An individual and its proposed scenario

The carrier-to-interference ratio (CIR or C/I), expressed in dB, is the ratio between the average received modulated signal power (i.e., $PR_{i,j}$) and the sum of co-channel interference power received from other transmitters (I_{Total}) [17]. CIR value can also be used as a deciding factor in the channel allocation to UTs [18]. The CIR perceived in BS_j is calculated based on the following expression:

$$CIR_j(dB) = PR_{i,j} - I_{Total} \tag{2}$$

Where, $PR_{i,j}$ is the received power from UT_i to BS_j. I_{Total} is the total interference caused by UTs using the same channel as UT_i (interference co-channel due to spectrum sharing).

From Equation 2, the total interference I_{Total} can be determined as follows:

$$I_{Total}(dB) = \sum_{m \in \varphi} PR_{m,j} \tag{3}$$

where $PR_{m,j}$ is the received power from UT_m to BS_j. m refers to the index of interfering transmitters that have been allocated to the same channel as UT_i. φ is the set of UT_m using the same channel.

The received power $PR_{i,j}$ in dB from UT_i to BS_j is determined as follows:

$$PR_{i,j}(dB) = PT_i - PL_{i,j} \quad (4)$$

where PT_i is the transmission power of the UT_i (uplink). $PL_{i,j}$ is the path loss expressed in dB. It represents the power reduction (attenuation) of the signal as it propagates through space between UT_i and BS_j. The path loss can be calculated using Hata's model for the urban area [19], which is specified as follows:

$$PL_{i,j}(dB) = A + B\, log_{10}(d_{i,j}) \quad (5)$$

where:

$$A = 69.55 + 26.16\, log_{10}(f_c) - 13.82\, log_{10}(h_j) - a(h_i) \quad (6)$$

$$B = 44.9 - 6.55\, log_{10}(h_j) \quad (7)$$

f_c is the carrier frequency, h_j is the BS antenna height, and h_i is the UT antenna height. We set A=50 and B=40, as did [20].

From Equation 3, the received power $PR_{m,j}$ from UT_m to BS_j is:

$$PR_{m,j}(dB) = PT_m - PL_{m,j} \quad (8)$$

where PT_m is the transmission power of the UT_m (uplink). $PL_{m,j}$ is the path loss between UT_m and BS_j given by:

$$PL_{m,j}(dB) = A + B\, log_{10}(d_{m,j}) \quad (9)$$

Fig. 4 depicts the CIR metric in BS_1. When BS_2 is switched off, its UTs are reassigned to neighbor BS_1. Then, each UT in BS_2 computes the received power in the channel that BS_1 allocates. This is shown in UT_6. Its interfering signals are those from UT_2 and UT_3, due to the fact that they are using the same channel as UT_6.

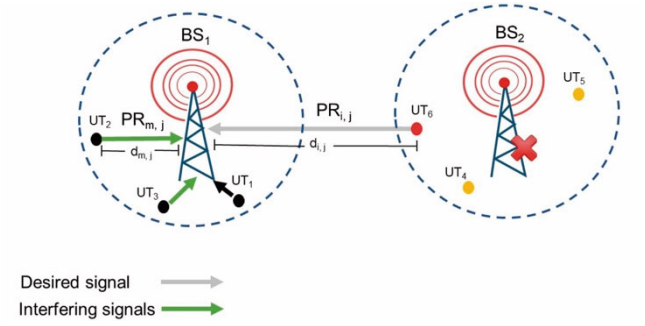

Figure 4
CIR metric in BS_1

The following optimization model accounts for the problem of optimizing the resources of a BS that is turned on according to the reassignment success of UTs. In order to obtain the minimum number of active BSs needed in a cellular network, the objective function is defined in Equation 10:

$$Minimize \sum_{j=1}^{J} SBS_j \tag{10}$$

As shown in Equation 10, the solution is an SBS vector that determines the lowest number of BSs to keep turned on. On the other hand, for a solution to be considered feasible, it must comply with the following restrictions:

$$\sum_{j=1}^{J} RBU_{j,i} = 1 \tag{11}$$

$$On_j = 1 \tag{12}$$

$$d_{i,j} \leq D \tag{13}$$

$$\sum_{i=1}^{I} RBU_{j,i} \leq MTU \tag{14}$$

$$CIR_j \geq \alpha \tag{15}$$

$$PI_k \leq \beta \tag{16}$$

Restriction (11) limits a UT_i to a single BS_j. Restriction (12) establishes that only active BSs may provide service to the reassigned UT; this is represented by the equation $On_j = 1$ if restriction (11) > 0 and $SBS_j = 1$, otherwise $On_j = 0$. The distance between a BS_j and a UT_i to which service is provided is limited in restriction (13), where it cannot be greater than the coverage radius threshold D of the BS_j. In (14), a BS is required not to exceed the maximum number of mobile users that it can service. The value of CIR_j represents interference perceived by a BS_j ,and it must be greater or equal to a threshold as set in (15). Lastly, the percentage of mobile users without service when the BSs are turned off in the k^{th} SBS vector must be lower than the threshold complying with (16). This percentage is the value of PI.

4 Algorithm for Base Station Deactivation and Mobile User Reassignment

The procedure to find the minimum set of active BSs to maintain service for at least 99% of UTs is described below. In STEP 1, the initial scenario is built: BSs have a fixed location, whereas the UTs within the coverage area are randomly deployed.

Each mobile user UT_i is assigned to a BS_j if it meets the following conditions: (1) the total number of UTs served by the BS_j is less than MTU and (2) the distance $d_{i,j}$ is less than or equal to D. The least used channel of the BS_j is assigned to each UT_i and the identifier of that channel is stored in vector CU in position i. According to the spectrum sharing technique, two or more UTs can share the same channel [16].

In STEP 2, binary values are randomly set for each element of the SBS vectors (individuals) that make up the population. NS identifies the size of the population.

In STEP 3, the k^{th} SBS vector k^{th} individual is evaluated. A_k represents its fitness. The following actions are carried out in this step:

1. Identify turned-off BS_j i.e. the elements of the SBS vector where $SBS_j = 0$
2. Based on the initial scenario, reassign to an active BS the UTs associated with the BS that is turned off in the k^{th} SBS vector, thereby complying with the conditions shown in Equations (11-15)
3. Calculate PI_k by applying Equation (17). PI_k is the percentage of UTs in the network that are out of service, i.e., those UTs that could not be reassigned to any of the active BSs in the k^{th} SBS vector. A UT_i is considered without service if all the cells in column i in the RBU matrix have a value equal to 0

$$PI = \left(I - \sum_{j=1}^{J} \sum_{i=1}^{I} RBU_{j,i} \right) * 100 \Big/ I \tag{17}$$

4. Evaluate the sum in Equation (10) to obtain the value of A_k, counting the elements of the k^{th} SBS vector, where $SBS_j = 1$

Input: Population size, crossover probability, mutation probability, and number of iterations. Total number of UTs, the maximum number of UTs per macro-cell, the transmission power of UTs, interruption probability threshold, number of femto-cells, width and height of the terrain, coverage radius for macro-cell/femto-cell, number of channels per macro-cell/femto-cell, and maximum number of UTs per femto-cell.

Output: The lowest number of BSs turned on and UTs reassignment.

BUILD initial scenario
INITIALIZE population with random individuals
EVALUATE each individual in Equations (10) to (16)
repeat
    SELECT two parents by using tournament selection
    RECOMBINE pairs of parents
    MUTATE the two-resulting offspring
    EVALUATE parents and offspring in Equations (10) to (16)
    SELECT the two best individuals out of the two parents and two offspring. Call those best individuals, *best1* and *best2*

10: REPLACE parents with *best1* and *best2* respectively
11: **until** *Number_of_cycles* < *Total_number_of_cycles*
12: SELECT the fittest individual from the population

If the PI_k value exceeds the threshold established in Equation (16), the k^{th} SBS vector will be penalized. That is, its fitness value will increase based on a penalty function. The k^{th} feasible vector can have a maximum value of $A_k = J$ (all active BSs). Therefore, the infeasible or penalized vectors will be added ($J + 1$) to their fitness value.

Each SBS vector corresponds to an RBU matrix that shows the reassignment of UTs to the active BS, the UTs without service (if they exist), and the deactivated BSs.

In STEP 4, the GA performs a cycle which is the process of selecting parents, crossing them, mutating offspring, and replacing parents.

In STEP 5, two parents are selected employing the tournament technique [12]. Two individuals are randomly picked, and the winner of these two individuals is selected as a parent (the individual with the lowest value of A_k). The process is then repeated (to generate a total of two parents).

Then, in STEP 6, a random number is generated within [0, 1], which is compared to the Crossover Probability (PC). If this random number is less than or equal to PC, two new individuals (offspring) are generated with a combination of the bits or elements of the parents. Specifically, we apply two-point crossover [12], where *c1* and *c2* are integers ranging from 1 to *J*.

In STEP 7, some bits of the offspring are mutated. A mutation is the inverse value of the bit. To decide which bits are to be mutated, a random number within the range of [0, 1] is generated for each element of the offspring. If this value is less than the mutation probability (PrM), the bit changes.

Once the offspring have been mutated, the algorithm proceeds in STEP 8 to evaluate these individuals and the parents. It also applies the four actions mentioned in STEP 3.

In STEP 9, the A_k values of the two parents and two offspring are compared. If those individuals are feasible solutions, the two individuals with the best fitness value are *best1* and *best2* [21]. Otherwise, if infeasible individuals are compared, the two individuals with the worst fitness value (the highest) are chosen to survive. Those two individuals are also called *best1* and *best2*. The replacement strategy applied when comparing infeasible individuals is another contribution from the present work.

In STEP 10, *best1* and *best2* are inserted into the population, replacing the parents.

There are different stop conditions for a GA with a steady-state population model. For example, if the optimal solution is known in the problem, the algorithm can be forced to perform the necessary cycles to find that solution or one very close to it.

In the case of a GA with a steady-state population model, a stop condition may be to carry out the necessary cycles so that all individuals in the population are replaced at least once by their offspring. Given that the optimal solution to the proposed problem is unknown and changing all the individuals could require too many cycles, the stop condition in the present work is a certain number of cycles, as mentioned in STEP 11.

STEP 12 is the result of STEPS 1 through 11. It determines the lowest number of required active BSs and reassigns those UTs whose BSs have been deactivated.

5 Results

Table 1 shows the values of the simulation parameters that were maintained in all the experiments carried out in this research. The aim was to simulate an LTE network. For this reason, parameters such as the transmission power of UTs were defined based on the work in [22].

The DeJong configuration presented in [23] was initially used regarding the GA parameters. It is a standard for many GAs, and this parameter combination has been found to work better for optimizing a function than many other parameter combinations.

However, to reduce the probability that the initial population is composed only of infeasible solutions, we increased the population size from 50 to 100 individuals. We also increased the number of cycles from 1000 to 2000, because we had observed decreases in fitness after cycle 1000. Finally, the GA parameters used for this specific problem are shown in Table 2.

A sensitivity analysis was carried out to observe the impact of the CIR variations on the PI values. In other words, we tried to figure out the trade-off between CIR and PI to achieve a service percentage of 99%.

Table 1
Simulation parameters

Parameter	Value
Transmission Power of UT	-40 dB
Area	25000000 m^2
Coverage radius D for macro/femto	1500 m/750 m
Number of channels per BS	10 channels
Interruption probability threshold	1%
Maximum number of users per macro-cell	150 UTs
Maximum number of users per femto-cell	75 UTs

Table 2
Parameters used in GA

Parameter	Value
Population size	100
Crossover probability	0.6
Mutation probability	0.001
Number of cycles	2000

Table 3 describes the characteristics of the experiments performed for sensitivity analysis. Regarding the type of network, two cases were considered: (i) a homogeneous network (only macro-cells) and (ii) a heterogeneous network (macro-cells and femto-cells). Two traffic statuses were established, a low one where there were only 500 UTs, and a high one where 1000 UTs were located within the cellular network. In each of the 12 experiments, the algorithm was run 50 times. As reported in [7], heterogeneous networks were used for the femto-cells to support the macro-cells in high-traffic status. The femto-cells were then deactivated at a low-traffic level.

Table 3
Parameters used in experiments for analyzing PI

Experiment	I	Type of network	Number of macrocells	Number of femtocells	CIR threshold (dB)
1	500	Homogeneous	10	-	3
2	1000	Homogeneous	10	-	3
3	500	Homogeneous	10	-	7
4	1000	Homogeneous	10	-	7
5	500	Homogeneous	10	-	14
6	1000	Homogeneous	10	-	14
7	500	Heterogeneous	5	5	3
8	1000	Heterogeneous	5	5	3
9	500	Heterogeneous	5	5	7
10	1000	Heterogeneous	5	5	7
11	500	Heterogeneous	5	5	14
12	1000	Heterogeneous	5	5	14

Table 4 shows the average fitness value for each experiment, the standard deviation, the lowest number of active BSs found in the best performance (best fitness obtained), and the highest number of activated BSs (worst fitness obtained). In addition, for experiments with heterogeneous networks, the fourth column (best found) specifies the number of activated macro-BSs (indicated by the letter M) and the number of activated femto-BSs (indicated by the letter F).

The data in Table 4 demonstrate that when the system network presents a low traffic status (experiments 1, 3, 5, 7, 9, and 11), the algorithm decides to turn off more BSs.

On average, these experiments kept around 6 BSs turned on. On the other hand, when the system network has high traffic (the remaining six experiments), the algorithm turns off fewer BSs. The latter experiments left about seven turned-on.

When the average number of active BSs is contrasted with the CIR threshold, it is clear that the higher the CIR threshold (14 dB) and traffic, the more BSs are turned on to maintain only 1% (PI threshold) or less of UTs without service.

Table 4
Fitness or number of activated BSs per experiment

Experiment	Average fitness	Standard deviation	The best found	The worst found
1	6.72	0.54	6	8
2	8.02	0.14	8	9
3	7.04	0.57	6	8
4	8.08	0.27	8	9
5	7.28	0.61	6	8
6	8.3	0.51	8	10
7	6.8	0.61	4 M + 2 F = 6	8
8	9.46	0.50	5 M + 4 F = 9	10
9	6.78	0.71	4 M + 1 F = 5	8
10	9.44	0.50	5 M + 4 F = 9	10
11	7.1	0.50	4 M + 2 F = 6	9
12	9.6	0.49	5 M + 4 F = 9	10

In each of the 12 experiments, when determining if a BS would remain active, the algorithm considers the BS's location concerning UTs. When a BS is centered in or close to an area with many UTs, and no other BS covering the majority of the UTs, the algorithm will likely keep the BS active. For example, the cellular network system in Experiment 1, shown in Fig. 5. The uppercase letters represent the macro-BSs, and the ones inside a red square represent the deactivated macro-BSs. It can be seen that J macro-BS is one of the farthest from the rest of the BSs, which makes it the only one that can cover certain UTs in its area. We observe that some BSs remain active in all the experiments, such as the J macro-BS. In contrast, macro-BSs A and D are chosen interchangeably in some experiments to cover the same area.

The above observations affirm that the algorithm prefers a BS with higher capacities. However, it cannot be ignored that the BSs are also chosen for the suitability of their locations. There have been cases where a femto-BS, situated in an important area to serve certain UTs, remained turned on even when traffic was low. Take, for example, the case of BS J. In experiments with heterogeneous networks, it became a femto-BS and was activated. The same happened in experiments 7 and 11.

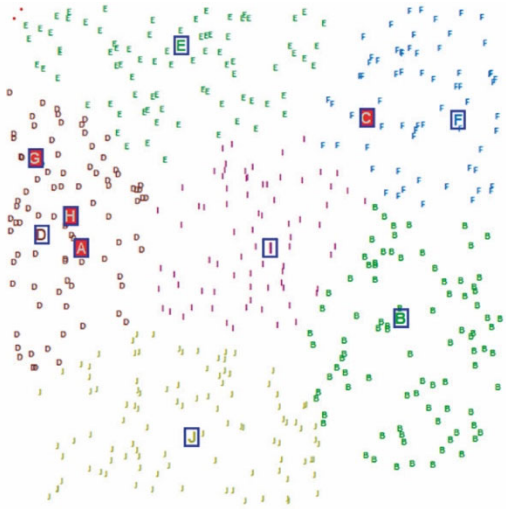

Figure 5
Cellular network system from experiment 1 after executing the algorithm

A completely different situation is seen when comparing the two types of networks in experiments with high traffic situations. Here the difference in fitness is very marked. For example, in Experiment 2, the average number of activated BSs was 8.02, whereas, in Experiment 8, the average was 9.46. We can infer that, in high traffic situations, the algorithm has to leave more BSs turned on when dealing with heterogeneous networks and fewer with homogeneous networks. This is due to the macro-BSs covering a larger area and a more significant number of UTs than the femto-BSs. For this reason, when the cellular network system is composed of femto-BSs and has high traffic, the algorithm is forced to keep more BSs active.

In terms of convergence, most experiments with low traffic status showed a trajectory similar to that shown in Fig. 6. In this case, since there were relatively few UTs, the algorithm found feasible solutions in early cycles. In contrast, most of the experiments with high traffic showed a convergence similar to that of Fig. 7. In that case, there were more UTs, so it was more challenging for the algorithm to find feasible solutions in the initial population. That performance was also due to the penalty function and the replacement strategy used when comparing infeasible individuals. The two worst individuals carried over to the next cycle because of this replacement strategy. Its effect was to increase fitness in early cycles, but as the cycles ran their courses, fitness decreased, resulting in a feasible solution. The improvement or deterioration in fitness was also a function of traffic. Take the experiment in Fig. 7 as an example. It had 1000 UTs. Once a feasible solution was obtained, the algorithm made few changes to reach a solution with fewer active BSs. In contrast, most of the experiments with 500 UTs, the algorithm made more changes to find solutions with inferior fitness (see Fig. 6).

In each of the 12 experiments, the solutions provided by the algorithm maintained the PI at 1%. Under the conditions specified in each experiment, at least one solution was found with a minimum percentage of UTs without service.

Finally, we evaluated other GA variants to compare performance. The experiment consisted of a homogeneous network (20 BSs, 500 UTs, and $\alpha = 3$ dB). The experiments were carried out using GA with the generational model and GA with the generational model using elitism. Those GA variants have two-point crossover and bit-flipping mutation. Their parameters are the ones shown in Table 2. Each GA variant executed 30 runs. The results are reported in Table 5.

Table 5 shows that the GA with the steady-state model outperforms the other GA variants. Consequently, the GA with the steady-state model has robustness since it has the lowest variation.

Table 5

Comparison of GA variants

GA variant	Average fitness	Standard deviation	The best found
GA with the steady-state population model	8.46	0.63	7
GA with the generational model	8.8	2.07	8
GA with the generational model using elitism	9.53	0.81	8

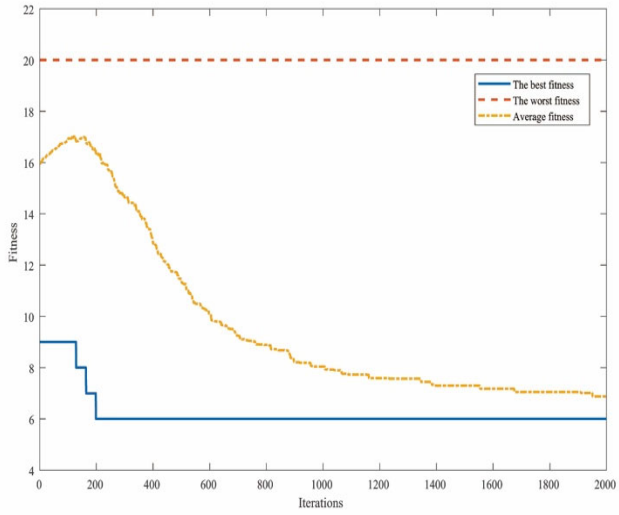

Figure 6

Convergence of the algorithm with low traffic (500 UTs)

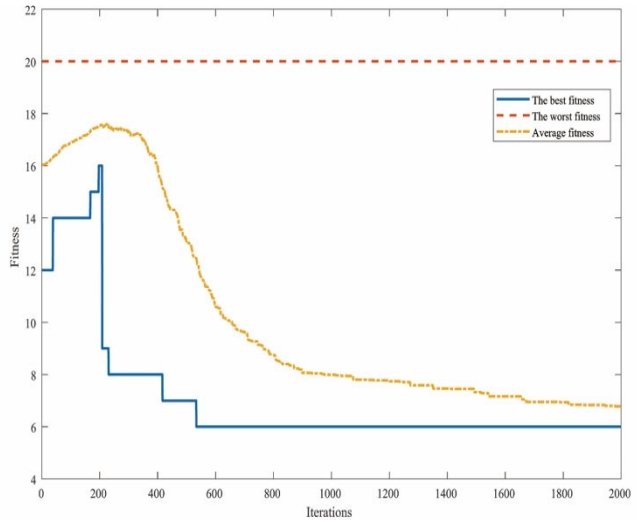

Figure 7
Convergence of the algorithm with high traffic (1000 UTs)

Conclusions

To take full advantage of dense 5G deployments, while still meeting the required QoS, sustainable management techniques are needed, to provide eco-friendly and cost-effective mobile architectures. A sustainable design of 5G systems includes sleep modes, that is, the capacity to turn off some of the BSs when the traffic load is low. Given the said context, we present our base station deactivation and user reassignment algorithm.

Based on the experiments carried out, our conclusions are as follows:

- One of the significant challenges of a deactivation and user-reassignment algorithm is to prevent the complete shutdown of all BSs in a network. For this reason, it is important to consider a mechanism that prevents infeasible solutions in the algorithm without compromising its performance.
- A deactivation algorithm based on a steady-state GA can successfully find a minimum set of active BSs because it shuts down 10-50% of the BSs present in a cellular network system and maintains service for at least 99% of users.
- In a cellular network, a reassignment process must be carried out to deactivate some BSs and maintain service for 99% of its UTs; not carrying out this process would leave up to 20% of UTs without service.
- An essential factor in the decision to deactivate a BS is the PI as the number of UTs without service in a cellular network. When considering this factor in regards to the optimization model, it is possible to switch off BSs according to the success of the UT reassignments and network traffic. When there is less traffic in the network, the number of active BSs is smaller.

- In all the proposed scenarios, even in those where the CIR threshold was equal to 14 dB, our proposed algorithm was able to find a solution where at least one BS was deactivated, and 99% of users were serviced.
- In heterogeneous networks, the algorithm deactivates more femto-BSs than macro-BSs when traffic is lower. This scheme supports the existing literature, which shows that the use of femto-BSs is more beneficial when the heterogeneous network presents high traffic status.

Going forward in our research, we will apply other metaheuristics to evaluate their performance in solving the problem addressed in this paper. We will pose the optimization model as a multi-objective problem, i.e., minimize the interruption probability and the number of active BSs. We plan to use the Page's Trend Test

Acknowledgements

Santiago-Manzano gratefully acknowledges the scholarship from CONACyT to pursue her graduate studies.

We thank Brianna Morin (Peace Corps Volunteer) for language editing that significantly improved the manuscript.

References

[1] Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper. 2020

[2] Alaca, F. et al.: A genetic algorithm based cell switch-off scheme for energy saving in dense cell deployments, 2012 IEEE Globecom Workshops, IEEE, 2012, pp. 63-68

[3] Nguyen, V. M., Kountouris, M.: Performance Limits of Network Densification, IEEE Journal on Selected Areas in Communications, 35 (6), 2017, pp. 1294-1308

[4] Son, K. et al.: Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks, IEEE journal on selected areas in communications, 29 (8), 2011, pp. 1525-1536

[5] Buzzi, S. et al.: A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead, IEEE Journal on Selected Areas in Communications, 34 (4), 2016, pp. 697-709

[6] Azzam, S. M., Elshabrawy, T.: Re-dimensioning number of active eNodeBs for green LTE networks using genetic algorithms, Proceedings of European Wireless 2015, 21th European Wireless Conference, 2015, pp. 1-6

[7] Alsharif, M. H. et al.: Survey of Green Radio Communications Networks: Techniques and Recent Advances, Journal of Computer Networks and Communications, 2013, 2013

[8] Piovesan, N. et al.: Energy sustainable paradigms and methods for future mobile networks: A survey. Computer Communications, 119, 2018, pp. 101-117

[9] Al-Kanj, L. et al.: Optimized joint cell planning and BS on/off switching for LTE networks, Wireless Communications and Mobile Computing, 16, 2016, pp. 1537-1555

[10] Kim, S. et al.: A joint algorithm for base station operation and user association in heterogeneous networks, IEEE Communications Letters, 17 (8), 2013, pp. 1552-1555

[11] Oh, E. et al.: Dynamic base station switching-on/off strategies for green cellular networks, IEEE transactions on wireless communications, 12 (5), 2013, pp. 2126-2136

[12] Eiben, A. E., Smith, J. E.: Introduction to Evolutionary Computing, Springer, Berlin, Heidelberg, 2015

[13] Niu, Z. et al.: Cell zooming for cost-efficient green cellular networks, IEEE Communications Magazine, 48 (11), 2010, pp. 74-79

[14] Yu, J. J.Q., Li, V. O. K.: Base station switching problem for green cellular networks with Social Spider Algorithm, 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 2338-2344

[15] Talbi, E.-G.: Metaheuristics: From Design to Implementation, John Wiley & Sons, 2009

[16] Martínez-Vargas, A., Andrade, Á. G.: Deployment analysis and optimization of heterogeneous networks under the spectrum underlay strategy, EURASIP Journal on Wireless Communications and Networking, 2015 (1), 2015, pp. 55

[17] Andrews, J. G. et al.: A primer on spatial modeling and analysis in wireless networks, IEEE Communications Magazine, 48 (11), 2010, pp. 156-163

[18] Katzela, I., Naghshineh, M.: Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey, IEEE Personal Communications, 3 (3), 1996, pp. 10-31

[19] Hata, M.: Empirical formula for propagation loss in land mobile radio services, IEEE Transactions on Vehicular Technology, 29 (3), 1980, pp. 317-325

[20] Brizuela, C. A., Gutiérrez, E.: An Experimental Comparison of Two Different Encoding Schemes for the Location of Base Stations in Cellular Networks, Applications of Evolutionary Computing (Editors: S. Cagnoni et al.), Springer, Berlin, Heidelberg, 2003, pp. 176-186

[21] Lozano, M. et al.: Replacement strategies to preserve useful diversity in steady-state genetic algorithms, Information Sciences, 178 (23), 2008, pp. 4421-4433

[22] Kumbhar, A. et al.: A Survey on Legacy and Emerging Technologies for Public Safety Communications, IEEE Communications Surveys Tutorials, 19 (1), 2017, pp. 97-124

[23] Sumathi, S., Paneerselvam, S.: Computational Intelligence Paradigms: Theory & Applications using MATLAB, CRC Press, 2019

What Is the Uncertainty of the Result of Data Processing: Fuzzy Analogue of the Central Limit Theorem

Julio C. Urenda[1,2], Olga Kosheleva[3], Shahnaz Shahbazova[4], and Vladik Kreinovich[2]

Departments of [1]Mathematical Sciences, [2]Computer Science, [3]Teacher Education
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA
jucrenda@utep.edu, olgak@utep.edu, vladik@utep.edu
[4]Azerbaijan Technical University, Baku, Azerbaijan, shahbazova@aztu.edu.az

Abstract: It is known that, due to the Central Limit Theorem, the probability distribution of the uncertainty of the result of data processing is, in general, close to Gaussian – or to a distribution from a somewhat more general class known as infinitely divisible. We show that a similar result holds in the fuzzy case: namely, the membership function describing the uncertainty of the result of data processing is, in general, close to Gaussian – or to a membership function from an explicitly described more general class.

Keywords: fuzzy logic; Central Limit Theorem; uncertainty

1 Introduction

1.1 Formulation of the problem

In the probabilistic approach to uncertainty, the most widely used probability distribution is normal (Gaussian). This fact has been empirically confirmed: for more than half of the measuring instruments, the probability distribution of the measurement error is close to Gaussian; see, e.g., [8, 9].

This fact also has a theoretical explanation: in most cases, the measurement error is caused by a joint effect of many small factors, and it is known that the distribution of the sum of a large number of small independent random variables is close to Gaussian. This theoretical explanation is known as the *Central Limit Theorem*; see, e.g., [12]. According to this theorem, when the number of summed variables increases, the probability distribution of their sum tends to Gaussian – this means exactly that as this number becomes large, the corresponding distribution is close to Gaussian.

In many practical situations, we do not know the corresponding distributions, all

we have is expert estimates for the approximation errors. These expert estimations are often described by using words from natural language like "small", "approximately", etc. A natural way to describe these estimates in precise computer-understandable terms is to use fuzzy logic – which was specifically designed for translating natural-language knowledge into such a precise form; see, e.g., [2, 3, 4, 6, 7, 13]. It is reasonable to expect that if we combine many such estimates, we should also get the resulting overall estimate in a specific form. What is this form? What is the resulting limit theorem – the analogue of the Central Limit Theorem? These are the questions that we study in this paper.

1.2 Outline of this paper

First, in Section 2, we analyze the general problem of estimating uncertainty of the result of data processing. In Section 3, we review the results related to the probabilistic case. In Section 4, we formulate the corresponding fuzzy case as a mathematical problem, and finally, in Section 5, we provide a solution to this problem.

2 Estimating Uncertainty of the Result of Data Processing: General Formulation of the Problem

2.1 What is data processing: a brief reminder

One of the main objectives of science and engineering is to predict what will happen in the world, and to come up with devices and techniques to make this future most beneficial for us.

The state of the world is characterized by the values of several quantities. For example, the state of the weather is described by temperature, humidity, wind speed, and wind direction. So, predicting the future state of the world means predicting the future values of these quantities.

Similarly, each device, each control strategy can be characterized by some numbers: e.g., if we control a car, then at each moment of time, we need to describe the value of the acceleration (if any is needed), and – if needed – the angular velocity with which the car is turning. So, coming up with the appropriate recommendations means estimating the values of the relevant quantities.

In both cases, we need to find an estimate $\widetilde{y}$ of each of the desired quantities y based on all available relevant information – i.e., based on the known estimates $\widetilde{x}_1,\ldots,\widetilde{x}_n$ of the corresponding quantities $x_1,\ldots,x_n$. The estimates $\widetilde{x}_i$ may come from measurements or they may come from experts.

In the following text, we will denote the algorithm used for estimating the desired quantity y by $\widetilde{y} = f(\widetilde{x}_1,\ldots,\widetilde{x}_n)$. Running these algorithms is what is usually called *data processing*.

2.2 How do we select data processing algorithms?

We select each data processing algorithm so as to best describe the relation between the corresponding quantities y and x_i. In other words, we select an algorithm f for

which, to the best of our knowledge, the actual values of these quantities satisfy the relation $y = f(x_1, \ldots, x_n)$.

2.3 Need to take uncertainty into account

Measurement results are never absolutely accurate. Expert estimates are usually even less accurate. In both cases, each available estimate $\widetilde{x}_i$ is, in general, different from the actual (unknown) value x_i of the corresponding quantity. In other words, there is, in general, a non-zero approximation error $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$. Because of this, the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of data processing is, in general, different from the actual value $y = f(x_1, \ldots, x_n)$: there is an uncertainty $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$.

For practical purposes, it is important to gauge this uncertainty. For example, if we are prospecting for oil, and we are estimating that a certain area contains 200 million tons, then our actions will depend on how accurate is this estimate. If it is 200 ± 50, then we should start exploiting this area right away, but if it is 200 ± 300, then maybe there is no oil at all, so it is better to perform further research before investing money in exploitation.

2.4 Data processing is often hierarchical

Data processing is often hierarchical, in the following sense. Instead of processing all the inputs right away, we divide them into groups – e.g., by time and/or by geographic locations. Then,

- first, we process inputs from each group, resulting in estimates for the combined quantities $z_1, \ldots, z_m$, and
- then, we use these estimates for z_j to estimate the desired value y.

This is how votes are counted in nation-wide elections, this is how data is often processed.

2.5 Possibility of linearization

In most practical situations, the approximation errors Δx_i are relatively small. In such cases, the terms which are quadratic in Δx_i can be safely ignored. For example, even if $\Delta x_i \approx 20\%$, the square of this number is 4%, which is much smaller. So, if we take into consideration that $x_i = \widetilde{x}_i - \Delta x_i$, expand the expression

$$\Delta y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(x_1, \ldots, x_n) = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n)$$

in Taylor series, and keep only terms linear in Δx_i in this expansion – while ignoring quadratic (and higher order) terms, we get an expression

$$\Delta y = c_1 \cdot \Delta x_1 + \ldots + c_n \cdot \Delta x_n, \tag{1}$$

where

$$c_i \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i}_{|(\widetilde{x}_1, \ldots, \widetilde{x}_n)}.$$

This is the main expression that we will use in our analysis of uncertainty of the result of data processing.

2.6 Linearization in the hierarchical case

In this case, in the first stage, we get

$$\Delta z_j = c_{j1} \cdot \Delta x_1 + \ldots + c_{jn} \cdot \Delta x_n, \tag{2}$$

where many of the coefficients c_{ji} – related to measurements x_i not from the group j – are 0s. Then, on the second stage, we get

$$\Delta y = c_1 \cdot \Delta z_1 + \ldots + c_m \cdot \Delta z_m. \tag{3}$$

3 Probabilistic Case: Brief Reminder

3.1 Central Limit Theorem: reminder

As we have mentioned, measurement errors are usually relatively small. Measurement errors corresponding to different measurements are usually independent. In practice, the value n is usually large. For example, to predict tomorrow's weather, we use thousands of recordings of weather conditions at different locations in different moments of time. To analyze an earthquake, we use thousands of values recorded by seismometers around it – or even, for a serious earthquake, all around the world. Thus, the formula (1) describes the sum of a large number of relatively small independent random variables. We have already mentioned earlier that, under reasonable conditions, the resulting distribution is close to Gaussian – this is what the Central Limit Theorem is about.

Thus, in the probabilistic case, we can conclude, with high confidence, that in many practical situations, the probability distribution of the uncertainty Δy with which we determine the result y of data processing is close to Gaussian.

3.2 Beyond the Central Limit Theorem

As we have commented, the convergence to the Gaussian distribution occurs under some reasonable conditions. What happens in the general case – when these conditions are not satisfied? To answer this question, let us take into account that data processing is often hierarchical.

If there is a limit theorem, according to which the probability distributions of the sums (1)–(3) are close to distributions of a certain type, then all variables Δz_j have distributions of this type, as well as the variable Δy. Thus, these limit distributions must have the property that a linear combination of thus distributed independent variables should have the distribution of exactly the same type.

In precise terms, when we say that we have a distribution of a certain type, we usually mean that there is a standard random variable ξ – e.g., normally distributed with mean 0 and standard deviation 1 – and all other distributions of this type has the same distribution as $d \cdot \xi$, for some constant d. In this case, if d_i is the value of the parameter d corresponding to Δz_j, then we can write Δz_j as $d_j \cdot \xi_j$, and the expression (3) as the sum

$$\Delta y = c_1 \cdot d_1 \cdot \xi_1 + \ldots + c_n \cdot d_n \cdot \xi_n,$$

i.e., equivalently, in the form

$$a_1 \cdot \xi_1 + \ldots + a_n \cdot \xi_n, \tag{4}$$

where we denoted $a_j \stackrel{\text{def}}{=} c_j \cdot d_j$.

In these terms, the above requirement states that each linear combination of identically distributed random variables ξ_j should have the same type of distribution, i.e., that for all possible values a_j, there should be the value a for which the sum (4) has the same probability distribution as $a \cdot \xi$.

Distributions with this property are known as *infinitely divisible*. Gaussian distribution clearly has this property, but there are other distributions with this property – e.g., Cauchy distribution, with the probability density function

$$f(x) = \frac{1}{\pi} \cdot \frac{1}{1+x^2}.$$

4 Fuzzy Case: Formulation of the Problem

4.1 What would a limit theorem mean in the fuzzy case: analysis of the problem

A similar argument can be repeated for the fuzzy case, when instead of probability distributions, we have membership functions – that describe, for each possible value x of the corresponding quantity, the degree (scaled to the interval $[0,1]$) to which this value is possible.

In this case, similarly to the probabilistic case, the existence of the limit theorem would mean that all linear combinations (1)–(3) are characterized by the same type of membership functions. This would mean, in particular, that if the quantities Δz_j are characterized by membership functions of this type, then their linear combination (3) is characterized by a membership function of the same type.

What does it mean "of the same type"? Similarly to the probabilistic case, a natural interpretation is that we should select one single membership function $\mu_0(x)$, and consider membership functions that describe quantities of the type $d \cdot \xi$, where the quantity ξ is described by a membership function $\mu_0(x)$.

What is the membership function of the quantity $d \cdot \xi$? To answer this question, let us recall that we can use different measuring units to describe the same value of the physical quantity. For example, to describe length, we can use meters, or we can use centimeters. If we replace the original measuring unit with a new one which is d times smaller, then all numerical values are multiplied by d: e.g., 2 meters becomes $2 \cdot 100 = 200$ centimeters. In general, the original numerical value x in the new scale is represented as $x' = d \cdot x$ – and, vice versa, the new value x' corresponds, in the original scale, to the value $x = x'/d$. Thus, if, in the original scale, the degree to which the value x is possible is $\mu_0(x)$, then the degree $\mu(x')$ to which the value x' on the new scale is equal to $\mu_0(x'/d)$.

So, quantities $d \cdot x$ are described by membership functions $\mu_0(x/d)$. In these terms, "membership function of the same type" means that we have a membership function of the type $\mu_0(x/d)$, i.e., for example, that the membership function of each quantity Δz_j is the same as the membership function of the product $d_j \cdot \xi_j$, where ξ_j has the membership function $\mu_0(x)$.

Thus, if there is a limit theorem, then, similarly to the probabilistic case, we conclude that:

- if we have several quantities $\xi_1, \ldots, \xi_m$ with the same membership function $\mu_0(x)$,
- then the membership function for a linear combination (4) should have the same membership function $\mu_0(x/a)$ as the quantity $a \cdot \xi$.

To describe this requirement in precise terms, let us recall how we can find the membership function corresponding to a linear combination (4).

4.2 How to find a membership function corresponding to a linear combination: Zadeh's extension principle

The value x is a possible value of the linear combination if there are some values ξ_j which are possible and whose linear combination (4) is equal to x. In general, "there exists" means that either this property holds for one combination of values ξ_j or for another combinations of values, etc.:

$$(\xi_1 \text{ is possible } \textit{and} \ldots \textit{and } \xi_n \text{ is possible } \textit{and } \sum_{j=1}^{m} a_j \cdot \xi_j = x) \textit{ or}$$
$$(\xi'_1 \text{ is possible } \textit{and} \ldots \textit{and } \xi'_n \text{ is possible } \textit{and } \sum_{j=1}^{m} a_j \cdot \xi'_j = x) \textit{ or}$$
$$\ldots$$

where "or" combines all tuples $(\xi_1, \ldots, \xi_m)$ for which $\sum_{j=1}^{m} a_j \cdot \xi_j = x$.

We know that all quantities ξ_j are described by the same membership function $\mu_0(x)$. This means that we know, for each value ξ_j, the degree to which this value is possible – this degree is equal to $\mu_0(\xi_j)$. According to the general fuzzy methodology, to find the degree of confidence in the above "and"-"or"-combination of such statements, we need to use appropriate "and"- and "or"-operations $f_\&(a,b)$ and $f_\vee(a,b)$ – also known as t-norms and t-conorms. Thus, the desired degree $\mu(x)$ has the form

$$f_\vee\left(f_\&\left(\mu_0(\xi_1), \ldots, \mu_0(\xi_m), d\left(\sum_{j=1}^{m} a_j \cdot \xi_j = x\right)\right), \right.$$
$$\left. f_\&\left(\mu_0(\xi'_1), \ldots, \mu_0(\xi'_m), d\left(\sum_{j=1}^{m} a_j \cdot \xi'_j = x\right)\right), \ldots\right).$$

Which "or"-operation should we choose? To make this choice, we need to take into account that there are infinitely many tuples ξ_j with the desired value x of the linear

combination, and thus, infinitely many terms combined by "or". For most "or"-operations (e.g., for $a+b-a\cdot b$), as we combine more and more statements, we will get closer and closer to 1. To avoid such a meaningless result, we need to use the only operation that does not increase the value – namely, the operation maximum. In this case, we get

$$\mu(x) = \max_{\xi_1,\ldots,\xi_m} f_{\&}\left(\mu_0(\xi_1),\ldots,\mu_0(\xi_m), d\left(\sum_{j=1}^{m} a_j\cdot\xi_j = x\right)\right).$$

Here, $d(S)$ is the degree to which the corresponding statement is true. In our case, the statement $\sum\limits_{j=1}^{m} a_j\cdot\xi_j = x$ is either true or false.

- If this statement is false, its degree is 0, so the whole combination has degree 0.
- If this statement is true, then its degree is 1, and this does not affect the result of the "and"-operation, since $f_{\&}(a,1)=a$.

Thus, we have

$$\mu(x) = \max_{\xi_j:\ \sum\limits_{j=1}^{m} a_j\cdot\xi_j = x} f_{\&}\left(\mu_0(\xi_1),\ldots,\mu_0(\xi_m)\right). \tag{5}$$

This formula – first derived by Zadeh – is known as *Zadeh's extension principle.*

4.3 Which "and"-operation should we use?

In the previous text, we showed which "or"-operation to use. A natural next question is: which "and"-operation should we use?

Some "and"-operations have the form

$$f_{\&}(a,b) = f^{-1}(f(a)\cdot f(b)) \tag{6}$$

for some strictly increasing function $f:[0,1]\to[0,1]$, where $f^{-1}(x)$ denotes the inverse function. Such "and"-operations are known as *strictly Archimedean*. It is known (see, e.g., [5]), that for every "and"-operation $t(a,b)$ and for every $\varepsilon>0$, there exists a strictly Archimedean "and"-operation $f_{\&}(a,b)$ for which

$$|t(a,b)-f_{\&}(a,b)|\le\varepsilon$$

for all a and b.

The whole idea of an "and"-operation is that the value $t(a,b)$ estimates the expert's degree of certainty in a statement $A\,\&\,B$ in a situation when we only know the expert's degrees of certainty a and b in statements A and B. Experts can estimate their degree of certainty only with some accuracy: we can usually distinguish between 7 and 8 on a 0-to-10 scale – which correspond to 0.7 and 0.8 – but it is doubtful that anyone can distinguish between degrees of certainty 0.70 and 0.71 – which correspond, for example, to marks 70 and 71 on a 0-to-100 scale. Since for sufficiently small ε, ε-close values are practically indistinguishable, in practice, it would

not make any difference if we use an ε-close strictly Archimedean "and"-operation instead of the original one $t(a,b)$.

So, from the practical viewpoint, it makes sense to assume that the actual "and"-operation used in the formula (5) is strictly Archimedean, i.e., that this "and"-operation has the form (6) for some strictly increasing function $f(x)$. In this case, the formula (5) takes the following form:

$$\mu(x) = \max_{\xi_j:\ \sum_{j=1}^{m} a_j\cdot\xi_j = x} f^{-1}(f(\mu_0(\xi_1))\cdot\ldots\cdot f(\mu_0(\xi_m))). \tag{7}$$

4.4 What does the limit property mean in this case

The above limit property means that the function $\mu(x)$ as described by the formula (7) also has the same form as the membership function $\mu_0(x)$, i.e., that it has the form $\mu(x) = \mu_0(x/d)$ for some value d.

So, the desired limit property takes the following form: *for each tuple $a_1,\ldots,a_m$, there exists a value d for which*

$$\mu_0(x/d) = \max_{\xi_j:\ \sum_{j=1}^{m} a_j\cdot\xi_j = x} f^{-1}(f(\mu_0(\xi_1))\cdot\ldots\cdot f(\mu_0(\xi_m))). \tag{7}$$

Let us call membership functions $\mu_0(x)$ satisfying this property *limit membership functions*. So, the question is: which membership functions are the limit ones?

5 Solution to the Problem: Description of All Possible Limit Membership Functions

5.1 Let us simplify the problem

In order to describe all possible limit membership functions, let us first simplify the above limit property as much as possible.

First, let us avoid the explicit use of the inverse function – since computing the inverse function is, in general, not easy. We can achieve this if we apply the function $f(x)$ to both side of the equality (7). If we take into account that this function is strictly increasing – so the largest (max) of its values is attained when x is the largest – then we can conclude that

$$f(\mu_0(x/d)) = \max_{\xi_j:\ \sum_{j=1}^{m} a_j\cdot\xi_j = x} (f(\mu_0(\xi_1))\cdot\ldots\cdot f(\mu_0(\xi_m))). \tag{8}$$

Now, let us make the constraint on ξ_j look simplest. For this purpose, let us denote by $v_j \stackrel{\text{def}}{=} a_j\cdot\xi_j$ the terms which are added in this constraint. In terms of these new

variables v_j, we have $\xi_j = v_j/a_j$. So, in terms of v_j, the formula (8) takes the following form:

$$f(\mu_0(x/d)) = \max_{v_j:\ \sum_{j=1}^{m} v_j = x} (f(\mu_0(v_1/a_1)) \cdot \ldots \cdot f(\mu_0(v_m/a_m))). \tag{9}$$

A further simplification can be done if we realize that in the formula (9), we only use the composition of the functions $f(x)$ and $\mu_0(x)$, but not the functions by themselves. To simplify the condition, let us therefore denote this composition by

$$\nu(x) \stackrel{\text{def}}{=} f(\mu_0(x)). \tag{10}$$

In terms of this new function, the formula (9) takes the following form:

$$\nu(x/d) = \max_{v_j:\ \sum_{j=1}^{m} v_j = x} (\nu(v_1/a_1) \cdot \ldots \cdot \nu(v_m/a_m)). \tag{11}$$

Next, we can replace multiplication – which is more complex than addition – with addition. There is a function specifically designed for this purpose – the logarithm function, for which $\ln(a \cdot b) = \ln(a) + \ln(b)$. So, instead of using $\mu(x)$, it makes sense to use $\ln(\nu(x))$. Since the logarithm is also a strictly increasing function, we conclude that

$$\ln(\nu(x/d)) = \max_{v_j:\ \sum_{j=1}^{m} v_j = x} (\ln(\nu(v_1/a_1)) + \ldots + \ln(\nu(v_m/a_m))). \tag{12}$$

A further minor simplification comes from the fact that since the values $\nu(x)$ are smaller than equal to 1, the logarithms of these values are negative (or 0). Since it is simpler to deal with positive numbers, let us multiply both sides of the formula (12) by -1. The corresponding operation $x \to -x$ is strictly decreasing, so it changes max to min. Thus, for the function

$$\ell(x) \stackrel{\text{def}}{=} -\ln(\nu(x)), \tag{13}$$

for which $\nu(x) = \exp(-\ell(x))$, we conclude that

$$\ell(x/d) = \min_{v_j:\ \sum_{j=1}^{m} v_j = x} (\ell(v_1/a_1) + \ldots + \ell(v_m/a_m)). \tag{14}$$

In particular, for $m = 2$, when $v_1 + v_2 = x$ and thus, $v_2 = x - v_1$, we conclude that

$$\ell(x/d) = \min_{v_1}(\ell(v_1/a_1) + \ell((x - v_1)/a_2)). \tag{15}$$

Now, we are ready to analyze this formula.

5.2 We have reduced our problem to a known problem in convex analysis

The above formula can be rewritten as

$$\ell_0(x) = \min_{v_1}(\ell_1(v_1) + \ell_2(x - v_1)), \tag{16}$$

where we denoted

$$\ell_0(x) \stackrel{\text{def}}{=} \ell(x/d), \;\; \ell_1(x) \stackrel{\text{def}}{=} \ell(x/a_1), \;\; \ell_2(x) \stackrel{\text{def}}{=} \ell(x/a_2). \tag{17}$$

The corresponding combination of the two function is known in *convex analysis* [10, 11], as the *infimal covolution*, or an *epi-sum*. It is usually denoted by

$$\ell_0 = \ell_1 \,\square\, \ell_2. \tag{18}$$

It is known that, under reasonable conditions, this formula can be further simplified if, instead of the original functions $\ell_i(x)$, we use their *Legendre-Fechnel transforms*

$$\ell_i^*(s) = \sup_x(s \cdot x - \ell_i(x)). \tag{19}$$

Namely, it is known [11] that the Legendre-Fechnel transform of the infimal convolution of two functions is equal to the sum of their Legendre-Fechnel transforms:

$$\ell_0^*(s) = \ell_1^*(s) + \ell_2^*(s). \tag{20}$$

5.3 Let us use this reduction

Let us describe the transform $\ell^*(s)$ of the function $\ell_i(x) = \ell(x/a_i)$ in terms of the Legendre-Fechner transform $F(s)$ of the function $\ell(x)$. Indeed, substituting the expression $\ell_i(x) = \ell(x/a_i)$ into the right-hand side of the formula (19), we conclude that

$$\ell_i^*(s) = \sup_x(s \cdot x - \ell(x/a_i)).$$

So, for the new variable $z \stackrel{\text{def}}{=} x/a_i$, for which $x = a_i \cdot z$, we conclude that

$$\ell_i^*(s) = \sup_z(s \cdot a_i \cdot z) - \ell(z)) = \sup_z((s \cdot a_i) \cdot z) - \ell(z)),$$

i.e., $\ell_i^*(s) = F(a_i \cdot s)$. Thus, the formula (20) takes the following form:

$$F(d \cdot s) = F(a_1 \cdot s) + F(a_2 \cdot s). \tag{21}$$

The requirement is that for every a_1 and a_2, there exists a value $d = d(a_1, a_2)$ for which the property (21) is satisfied. Differentiating both sides of this equality by a_2, we conclude that

$$s \cdot F'(a_2 \cdot s) = a \cdot s \cdot F'(d(a_1, a_2) \cdot s),$$

where we denoted

$$a \stackrel{\text{def}}{=} \frac{\partial d}{\partial a_2}_{|(a_1, a_2)}.$$

Dividing both sides by s, we conclude that

$$F'(a_2 \cdot s) = a(d, a_2) \cdot F'(c \cdot s).$$

In particular, for $a_2 = 1$, we conclude that $F'(s) = a(d,1) \cdot F'(d \cdot s)$, i.e., that

$$F'(d \cdot s) = A(d) \cdot F'(s),$$

where we denoted $A(d) \stackrel{\text{def}}{=} \dfrac{1}{a(d,1)}$. It is known (see, e.g., [1]) that every continuous solution to this functional equation has the form $F'(s) = b \cdot s^{\alpha}$. Integrating, we conclude that $F(s) = B \cdot s^{\beta} + C$ for some constants B, β, and C.

Substituting this formula into the condition (21), we conclude that $C = 0$ and thus, that $F(s) = B \cdot s^{\beta}$. It is known that if the Legendre-Fechnel transform of a function is a power law, then the function itself is a power law, so

$$\ell(x) = D \cdot x^{\gamma} \tag{23}$$

for some D and γ, and thus, that the function $\nu(x) = \exp(-\ell(x))$ has the form

$$\nu(x) = \exp(-D \cdot x^{\gamma}), \tag{24}$$

and thus, for $\mu(x) = f^{-1}(\nu(x))$, we have $\mu(x) = f^{-1}(\exp(-D \cdot x^{\gamma}))$.

5.4 Conclusion: fuzzy analogue of the Central Limit Theorem

In the probabilistic case, due to the Central Limit Theorem, the uncertainty of the result of data processing is described by a Gaussian distribution or, more generally, by an infinitely divisible distribution.

Similarly, for the membership function $\mu(\Delta y)$ describing the uncertainty of the result of data processing, we can make the following conclusion:

- when the "and"-operation is the algebraic product, then

$$\mu(\Delta y) = \exp(-D \cdot |\Delta y|^{\gamma}); \tag{25}$$

- in general, when the "and"-operation has the form

$$f_{\&}(a,b) = f^{-1}(f(a) \cdot f(b)),$$

then

$$\mu(\Delta y) = f^{-1}(\exp(-D \cdot |\Delta y|^{\gamma})). \tag{26}$$

Acknowledgements
This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478.

References

[1] J. Aczél and J. Dhombres: Functional Equations in Several Variables, Cambridge University Press, 2008.

[2] R. Belohlavek, J. W. Dauben, and G. J. Klir: Fuzzy Logic and Mathematics: A Historical Perspective, Oxford University Press, New York, 2017.

[3] G. Klir and B. Yuan: Fuzzy Sets and Fuzzy Logic, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[4] J. M. Mendel: Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions, Springer, Cham, Switzerland, 2017.

[5] H. T. Nguyen, V. Kreinovich, and P. Wojciechowski: Strict Archimedean t-Norms and t-Conorms as Universal Approximators, International Journal of Approximate Reasoning, 1998, Vol. 18, Nos. 3–4, pp. 239–249.

[6] H. T. Nguyen, C. L. Walker, and E. A. Walker: A First Course in Fuzzy Logic, Chapman and Hall/CRC, Boca Raton, Florida, 2019.

[7] V. Novák, I. Perfilieva, and J. Močkoř: Mathematical Principles of Fuzzy Logic, Kluwer, Boston, Dordrecht, 1999.

[8] P. Novitsky and I. Zograph: Errors Estimation for Measurements Results, Energoatomizdat, Leningrad, 1991 (in Russian).

[9] A. I. Orlov: How often are the observations normal?, Industrial Laboratory, 1991, Vol. 57, No. 7, pp. 770–772.

[10] A. Pownuk, V. Kreinovich, and S. Sriboonchitta: Fuzzy data processing beyond min t-norm, In: C. Berger-Vachon, A. M. Gil Lafuente, J. Kacprzyk, Y. Kondratenko, J. M. Merigo Lindahl, and C. Morabito (eds.): Complex Systems: Solutions and Challenges in Economics, Management, and Engineering, Springer Verlag, Cham, Switzerland, 2018, pp. 237–250.

[11] R. T. Rockafeller: Convex Analysis, Princeton University Press, Princeton, New Jersey, 1997.

[12] D. J. Sheskin: Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hall/CRC, Boca Raton, Florida, 2011.

[13] L. A. Zadeh: Fuzzy sets, Information and Control, 1965, Vol. 8, pp. 338–353.

An Input-weighted, Multi-Objective Evolutionary Fuzzy Classifier, for Alcohol Classification

Shahnaz N. Shahbazova[1] and Dursun Ekmekci[2,*]

[1] Department of Computer Technologies and cybersecurity, Azerbaijan Technical University, shahbazova@aztu.edu.az
Azerbaijan National Academy of Sciences, Institute of Control System, shahbazova@isi.az; Baku, Azerbaijan; ORCID ID: 0000-0002-9898-6829

[2] Department of Computer Engineering, Faculty of Engineering, Karabuk University, Karabuk, Turkey, E-mail: dekmekci@karabuk.edu.tr

* Corresponding Author; ORCID ID: 0000-0002-9830-7793

Abstract: The success of the evolutionary computational methods in scanning at problem's solution space and the ability to produce robust solutions, are important advantages for fuzzy systems, especially in terms of "interpretability" and "accuracy". Many techniques have been introduced for multi-objective evolutionary fuzzy classifiers by considering this advantage. However, these techniques are mostly fuzzy rule-based methods. In this study, instead of designing an optimal rule table or determining optimal rule weights, the inputs are weighted, and no rules are used. The average of the degrees of membership obtained with their Membership Function (MF) is calculated as the "input membership degree (μInp)" for each input. The μInps are then weighted, and a single coefficient is generated to be used for the output. With the output, results are obtained for different objective functions. The weights of the inputs and the MFs parameters of all variables (inputs and outputs) are optimized with NSGA-II. The performance of the method has been tested for alcohol classification. As a result, it has been proven that the method can generate designs that can classify at shallow error levels with different sensors at different gas concentrations. In addition, it has been observed that the proposed method produces more successful solutions for alcohol classification problems when compared to other MOEFC techniques.

Keywords: Multi-Objective Fuzzy Classifier; Multi-Objective Optimization; Input-Weighted Multi-Objective Fuzzy Classifier

1 Introduction

One of the main issues to be considered in Fuzzy Systems design is optimizing the balance of "interpretability" and "accuracy" which generally conflict with each other. Evolutionary computational methods have been proposed in many studies for this delicate balance. The success of Evolutionary Algorithms (EAs) for designing the architectures of single-output fuzzy systems has also inspired Multi-Objective Evolutionary Fuzzy Classifiers (MOEFCs). In this context, MOEFC has a hybrid structure that combines the approximate reasoning capability of fuzzy logic with the robust adaptation performance of EAs, for complex classification problems. Within the scope of MOEFCs, EAs are applied to Fuzzy Rule-Based Systems (FRBSs) for rule tuning, mining, selection, weighting and are applied to Fuzzy Inference Systems (FISs) for parameter tuning. EAs can also easily incorporate prior knowledge into the system [1]. During the evolutionary design process, models are widely used to approach classification problems as they are characterized by a good balance between their accuracy and their level of interpretability [2].

The two main components that determine the performance of a Fuzzy Classifier (FC) are the adequate structure and the determination of the parameters. While constructing the structure of an FC, choosing the adequate variables, assigning enough Membership Functions (MFs) for each variable, and designing a practical fuzzy rule table are essential for the model's performance. In addition to these tasks, setting the MFs' parameters will become highly complex due to its vast search space, especially when considering high-dimensional problems. This challenge in the FC design is examined in detail in [3]. To overcome this problem, although different heuristic techniques are suggested today, the Genetic Algorithm (GA) was primarily preferred in the first examples [4]-[5]. EA-based FCs are generally rule-based systems. Ishibuchi et al. [4] used a method to minimize the number of fuzzy rules on the one hand and increase accuracy on the other. Gorzalczany and Rudzinski [6] applied their proposed multi-objective GA method in the technical field of glass identification in forensic science as decision support. In [7], fuzzy sets are not tuned, but prior knowledge of the distribution of fuzzy sets is required. Ducange et al. [8] tested their proposed MOEFC method on two Internet traffic datasets obtained from real-world networks. They applied cross-validation and cross-testing on the datasets. In both cases, they achieved successful low complexity and high interpretability results. Pietari et al. [9] proposed a different approach for FRBS design. True positive and false positive rates were determined instead of the commonly used misclassification rate as accuracy measures. The model also has interpretability, which is then allowed to be adjusted. The method used the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [10] method to balance objectives.

The convergence performance of the model is low in approaches that randomly generate the initial population [11]. Also, some methods use aggregate fitness

functions [12] [13]. Vaishali et al. [14] aimed to improve the accuracy of existing diagnostic procedures in predicting Type 2 Diabetes. In the initialization phase, they selected the essential features with GA from the dataset they used and applied MOEFC on the features. In the method, they achieved the maximum rate of the classifier with the minimum number of rules.

When literature studies are examined, it is seen that MOEFC methods are mainly based on FRBSs. In most studies, the number of fuzzy rules and the resulting error were considered objectives for balancing accuracy and interoperability. Unlike the classical FRBSs, this study uses the weighted-input approach, thus eliminating the need for effective rule design or optimal rule weighting. Another advantage of the method is that it can provide information about the relative importance of the inputs for all objectives in the problem. The average of the membership degrees obtained with its MFs is calculated as "the input membership degree (μInp)" for each input. Then, μInps are weighted, and a single coefficient is produced to be used for the output. With the output, results are obtained for different objective functions. The weights of the inputs and the MFs parameters of all variables (inputs and outputs) are optimized with NSGA-II. The performance of the proposed method has been tested on the alcohol classification problem. Using five Quartz Crystal Microbalance (QCM) sensors with different structures, measurements have been obtained in environments with different gas concentrations. The objective is to design a fuzzy classifier that can classify five different types of alcohol by evaluating the measurements of a QCM sensor. In this context, the main idea of the study is to design a MOEFC that can make the best classification for all sensors. Experimental results have proven that the method can successfully classify five different types of alcohol with a single solution vector. In [15], a coding scheme using accuracy and diversity and an entropy-based diversity criterion are proposed in evolutionary multi-objective optimization algorithms for MOEFC.

The remainder of the paper is designed as follows: In Section 2, a background of the study is explained. First, the concept of Multi-Objective Optimization (MOO) is emphasized. Then, the NSGA-II method, which can be successfully applied to Multi-Objective Optimization Problems (MOOP), is explained with its main steps and basic procedures. Finally, the proposed method is introduced in the section. In Section 3, first, the experiments for alcohol classification and the data set designed according to the results of the experiments are described. Then, the implementation of the method to the problem is explained and finally, the results are shared and interpreted in detail. Section 4 concludes this work.

2 The Background of the Proposed Method

From the MOEFC perspective, the basic approach of MOO methods is to search for a set of non-dominated fuzzy systems with different trade-offs between accuracy and complexity. For an effective MOEFC design, accuracy maximization is as crucial as complexity minimization. Within the scope of the study, the MOEFC method, which aims for optimal classification by the same solution vectors, has been proposed for these conflicting objectives. The NSGA-II algorithm is preferred for parameter tuning of MFs, and optimal input weights in the proposed method. Therefore, this section examines the concept of MOO, and the NSGA-II algorithm is explained. Finally, the proposed method is introduced in detail.

2.1 Multi-Objective Optimization

The MOOP can be formally expressed as in [16]: finding an n-dimensional possible solution vector $x = (x_1, x_2, x_3, \ldots, x_n)^T$ of decision variables that will satisfy many constraints and optimizes the vector function $f(x)=[f_1(x), f_2(x), f_3(x), \ldots, f_r(x)]$ and $D \subseteq R^n$ is an n-dimensional bounded decision space. R represents the objectives. The constraints define the objective space $\mathcal{F}$, containing all the admissible solutions. Since it is challenging to optimize conflicting objectives simultaneously, a set of Pareto optimal solutions is generated instead of a single optimal solution. Pareto optimal solutions present objective function values of a multi-objective optimization model. None of the objective functions can be increased in value without decreasing some of the other objective values in this set of solutions [17].

Without loss of generality, this study adopts the following basic concepts of MOO:

- ***Pareto dominance:*** Feasible solutions $x \prec y$ if and only if $f_i(x) \prec f_i(y)$ $(\forall i=1, 2, 3, \ldots, m)$ and $f_j(x) \preceq f_j(y)$ $(\exists j \in \{1, 2, 3, \ldots, m\})$
- ***The Pareto optimal set (or non-dominated set)*** is defined as PS = {x ∈ D| x is Pareto optimal} and *the Pareto optimal front* is defined as $PF^* = \{f(x)| x \in PS\}$
- ***External archive:*** A solution matrix saves the non-dominated solution vectors achieved so far

Although many GA-based techniques have been developed for MOOP, NSGA-II is more advantageous than its counterparts in terms of computation time [18]. Deb et al. [10] showed that NSGA-II could produce more successful solutions than many other MOO techniques in finding an alternative set of solutions and converging to the actual Pareto-optimal set. Moreover, in their comprehensive survey on the controller tuning problem in intelligent control systems, Rodríguez-Molina et al. [19], emphasized that NSGA-II is the popular choice compared to

other meta-heuristic methods. Therefore, in this study, the NSGA-II method was preferred.

2.2 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

The first EA-based methods proposed for MOOP were generally developed based on GA [1]. NSGA [20] initially developed for real parameter optimization in multi-objective constrained optimization problems, is one of the first famous examples of these methods [2]. However, NSGA has been criticized for its high computational complexity, lack of elitism, and the necessity of determining the sharing parameter, and its improved versions are presented [21]. In this context, the NSGA-II [10] is a significantly revised version of NSGA. The NSGA-II includes three basic procedures: fast non-dominated sorting (for the entire population [14]), crowding distance assignment, and the main loop.

Formally, the NSGA-II can be briefly summarized as following steps [22].

Initialize solutions: Generating initial solutions considering the lower and upper bounds.

Non-dominated sorting: Sorting the initial solutions according to the criteria of non-domination.

Crowding distance: Once the sorting is complete, the crowding distance value is assigned to the front. Solutions are selected according to rank and crowding distance.

Selection: The selection of solutions is carried out using a binary tournament selection with the crowded-comparison operator ($\prec_n$).

Genetic operators: New solutions are produced by crossover and mutation operations.

Recombination and selection: Old and new solutions are combined, and the solutions to be used in the next cycle are determined by selection. Solution selection continues for each objective until the number of populations exceeds the number of solutions available.

2.3 Proposed Method: An Input-weighted Multi-Objective Evolutionary Fuzzy Classifier

MOEFCs are the techniques in which fuzzy approach and multi-objective EAs are hybridized. Therefore, in this section, the proposed method is introduced from the side of both main components.

2.3.1 Fuzzy Logic Side

The proposed MOEFC technique differs from the classical fuzzy logic system. In the method, all MFs of the input and output variables are of type *"Gaussian combination membership function (gauss2mf)"* [23]. Compared to other MFs, in many studies, better solutions have been obtained with the gauss2mf [24] [25].

gauss2mf calculates the membership degrees using a combination of two Gaussian MFs given in (1).

$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{1}$$

where σ represents the standard deviation, and c represents the mean for the Gaussian function. Membership value is computed for x.

gauss2mf can be used on the MATLAB platform, as given in (2) [26].

$$y = \text{gauss2mf}(x, [\sigma_1\ c_1\ \sigma_2\ c_2]) \tag{2}$$

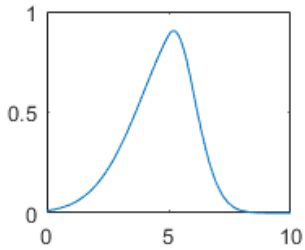

Figure 1
gauss2mf with the parameters σ_1=2, c_1=6, σ_2=1, c_2=5

Figure 1 shows the gauss2mf plotted with parameters σ_1=2, c_1=6, σ_2=1, c_2=5. Each Gaussian function defines the shape of one side of the MF. The left curve is drawn using the parameters σ_1 and c_1 for (1). The parameters σ_2 and c_2 are used for (3), and the right curve is drawn.

$$f(x;\sigma,c) = 1 - e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{3}$$

In addition, *"the input membership degree (μInp)"* is determined for each input. The membership value of *μInp1* with n MFs for x is computed by (4).

$$\mu Inp1(x) = (\sum_{i=1}^{n} \mu Inp1.MF_i(x)) / n \tag{4}$$

Then, all the inputs are weighted. Using these weighted inputs, the coefficient z is calculated with (5).

$$z = (\sum_{i=1} (\mu Inp_i * w_i)) / (\sum_{i=1} w_i) \tag{5}$$

In (5), w_i is the randomly assigned weight for μInp_i. The output is the average of the membership values calculated with the MFs of the output variable for the coefficient z. For output with n MFs, the μOut is calculated by (6).

$$\mu Out(z) = (\sum_{i=1}^{n} \mu Out.MF_i(z)) / n \tag{6}$$

In the fuzzy system design described, the NSGA-II method is used to optimize the parameters of the MFs of all variables, the weights of the inputs, and the output that determines the system results for different objectives.

2.3.2 NSGA-II Side

In the proposed method, the number of weights to be optimized is equal to the number of inputs. MFs in both input and output variables are of the gauss2mf type. As shown in (2), gauss2mf is a function that has 4 parameters. Accordingly, the number of parameters to be optimized for MFs will be 4 times the total number of MFs. Thus, the number of dimensions (D) in each solution vector is calculated with (7).

$$D = count(Input) + 4 * count(MFs\ of\ \text{var}iables) \tag{7}$$

The output takes values in the range [0, 1]. In this context, lower bound and upper bound points are determined in the range of [0, 1] for each class. In the proposed method, the aim is to bring the outputs closer to the center of the targeted class. Therefore, for each class, the center point must be calculated. Table 1 shows the classes' lower bound, upper bound, and center points for a classification problem with c classes.

Table 1
The lower bounds, upper bounds, and center points calculated for c classes

	Class 1	**Class 2**	**…**	**Class m**
Lover bound	0	1/m		m-1/m
Upper bound	1/m	2/m		1
Center	1/(2*m)	3/(2*m)		(2*m-1)/(2*m)

The absolute value of the difference between the output produced by the system and the center point of the targeted class is measured as the error (e), as given in (8).

$$e = \left| Out - center \right| \tag{8}$$

The errors are calculated for all patterns in the data set, and the total error (E) is determined by (9).

$$E = \sum_{i=1} e_i \tag{9}$$

The objective function of the proposed method, given in (10), is to minimize the classification error obtained for each objective with the same solution vector.

$$f(E) = \min(f_1(E), f_2(E), f_3(E), ..., f_r(E)) \tag{10}$$

3 Experimental Study

The performance of the proposed method is tested on a dataset used in [27] and shared in the UCI database, designed with data from five different sensors for the alcohol classification problem and can be found at

https://archive.ics.uci.edu/ml/datasets/Alcohol+QCM+Sensor+Dataset

The method was coded in the MATLAB R2017b platform and run on a computer having the Intel(R) Core (TM) i7-4710MQ 2.50 GHz processor with 8 GB RAM and Windows 8 operating system.

This section introduces the selected MOOP, and the experiments for the dataset used are explained. Then, the proposed MOEFC method implementation to the problem is presented, and the obtained results are discussed in detail.

3.1 Selected MOOP and Dataset

Within the scope of the study, the alcohol classification problem is selected as an example of MOOP. The problem is one of the popular classification problems, which has been studied for years and offers solutions with different techniques.

3.1.1 Alcohol Classification Problem

Recognition and classification of chemical compounds play an essential role in determining the compound's usage areas and harmful effects. In this regard, alcohols are many chemical compounds in the cosmetic and hygiene industry [27]. One of the sensors that can detect types of alcohol is a Quartz Crystal Microbalance (QCM) [28]. The QCM is essentially an electromechanical oscillator and has the characteristics of a sensitive piezoelectric effect [29]. It is widely used as a gas sensor in cases where chemicals in gases have different densities according to their types. However, precise detection in a sensor cannot be classified all at once [30]. Therefore, using these sensors with artificial

intelligence techniques is less costly. Thus, an informed decision about many chemical products can be made automatically.

3.1.2 Data Background

In this study, 5 different types of alcohol are classified as 1-octanol, 1-propanol, 2-butanol, 2-propanol, and 1-isobutanol, with 5 different QCM sensors, as in [27]. Each of the QCM sensors has two different channels: the channel including "molecularly imprinted polymers (MIP)" and the channel including "nanoparticles (NP)". The MIP and NP ratios used in the sensors are: 1-1, 1-0, 1-0.5, 1-2, and 0-1, respectively. The gas sample is passed through each sensor at five different air-gas concentrations, and the measurements obtained are saved in the data set. The ratio of air and gas concentrations in ml is presented in Table 2.

Table 2
Air-gas concentrations in experiments

	Air ratio	**Gas ratio**
1	0.799	0.201
2	0.700	0.300
3	0.600	0.400
4	0.501	0.499
5	0.400	0.600

3.1.3 Dataset Design

25 experiments were performed with each QCM sensor at the specified MIP and NP channel ratios and in the environments presented in Table 2, that is, for a total of 50 different scenarios. Therefore, there are 1250 samples in the data set.

The data set values obtained for each scenario are normalized in the range to [0 1] with (11). x_i represents the number to be normalized, x_{min} and x_{max} represent the minimum and maximum values in the respective scenario, respectively.

$$x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{11}$$

60% of the samples (15 samples) in each scenario were used for training and 40% (10 samples) for testing. These training and testing samples are selected randomly in each scenario.

3.2 Implementation of the Proposed Method to the Problem

In the main structure of the system, measurements in different gas concentrations are included as inputs to the system, and the obtained output membership degree

(μOut) is used for classification with each sensor. In the study, equal numbers of MFs are used in the variables (1 to 4). Figure 2 illustrates a design for training the proposed method with 2 MFs in each variable.

When equations (3), (4), and (5) are used with the design parameters given in Figure 2, the μOut of the system can be calculated. The produced μOut value is evaluated separately for each sensor. The system's training aims to get the μOut values closer to the class centers. Thus, the main objective is to design a model that obtains minimum error for all sensors [31].

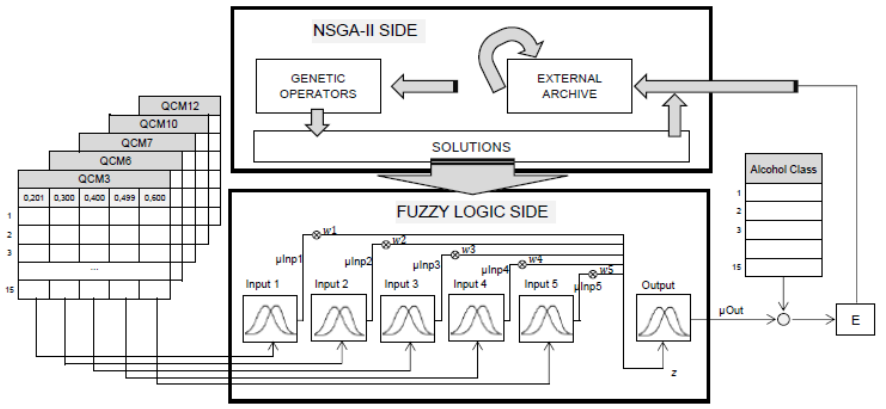

Figure 2

Proposed MOEFC model that has 2 MFs in its variables

Since the system will produce μOut value in the range of [0, 1], the lower and upper bounds and center points are assigned in the range of [0, 1] for alcohol classes. Accordingly, the determined values are shown in Table 3.

Table 3

The lower bounds, upper bounds, and center points assigned for the alcohol classification

	1-octanol	**1-propanol**	**2-butanol**	**2-propanol**	**1-isobutanol**
Lover bound	0	0.2	0.4	0.6	0.8
Upper bound	0.2	0.4	0.6	0.8	1
Center	0.1	0.3	0.5	0.7	0.9

The performance of the system is determined by reference to the values in Table 3. Accordingly, the error (E) on n samples is calculated by (12) for each sensor.

$$E = \sum_{i=1}^{n} | \mu Out_i - Center_i | \tag{12}$$

In terms of genetic operators, the length of each artificial chromosome is determined by selected variable numbers. The weights to be assigned are equal to the number of inputs, and considering Eq. (2), 4 parameters are required for each MF. Accordingly, for the 1, 2, 3, and 4 MFs numbers used in the experiments,

solution vectors with 29, 53, 77, and 101 items are required, respectively. The artificial chromosome structure designed for the system shown in Figure 2 is presented in Figure 3.

As seen in Figure 3, the first 5 items in the solution vector are the weights assigned to the inputs randomly. The following 4 items are the parameters set to the gauss2mf (σ_1, c_1, σ_2, c_2) type MF of Inp1. Since each variable has 2 MFs, 8 parameters are required for all variables.

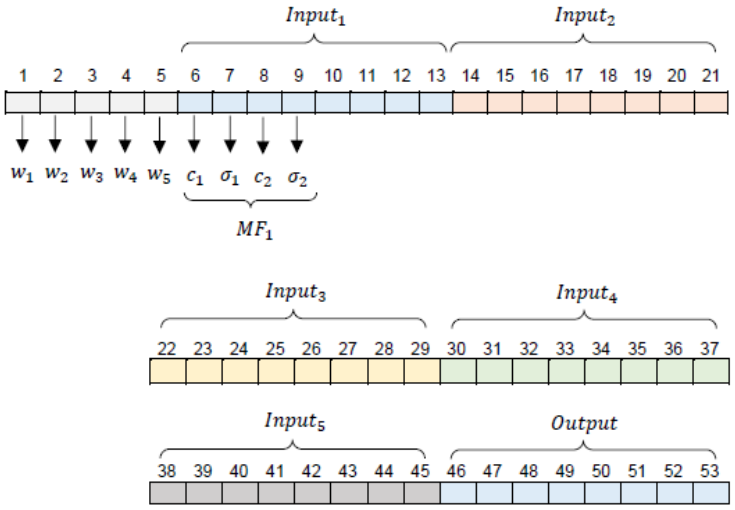

Figure 3
Detail of a solution vector for the model that has 2 MFs in its variables

The parameter settings for the NSGA-II are shown in Table 4. The maximum cycle number (MCN) is set to 10000 in each trial. The algorithm has been run 3 times independently for each scenario.

Table 4
Parameter setting for the NSGA-II

Parameter	Value
Population size	50
Crossover fraction	0.8
Mutation fraction	0.1
Pareto front population	50

3.3 Results and Discussion

The results obtained with the proposed MOEFC method are shared and discussed in detail in this section. The results are evaluated in 4 categories based on the MF numbers used.

The results obtained in the experiments are presented in Figures 4-7, with graphs drawn for different goals. Graphs "a" in the figures: error values calculated based on sensors at the end of the training process, graphs "b": number of misclassifications for training data, based on sensors, graphs "c": weights assigned to inputs, and graphs "d": number of misclassifications obtained for test samples, based on sensors.

3.3.1 Scenarios That Have 1 MF in Each Variable

The proposed method's results by using only one MF in each variable are examined. This experiment is essential to analyze the interpretative ability of the technique.

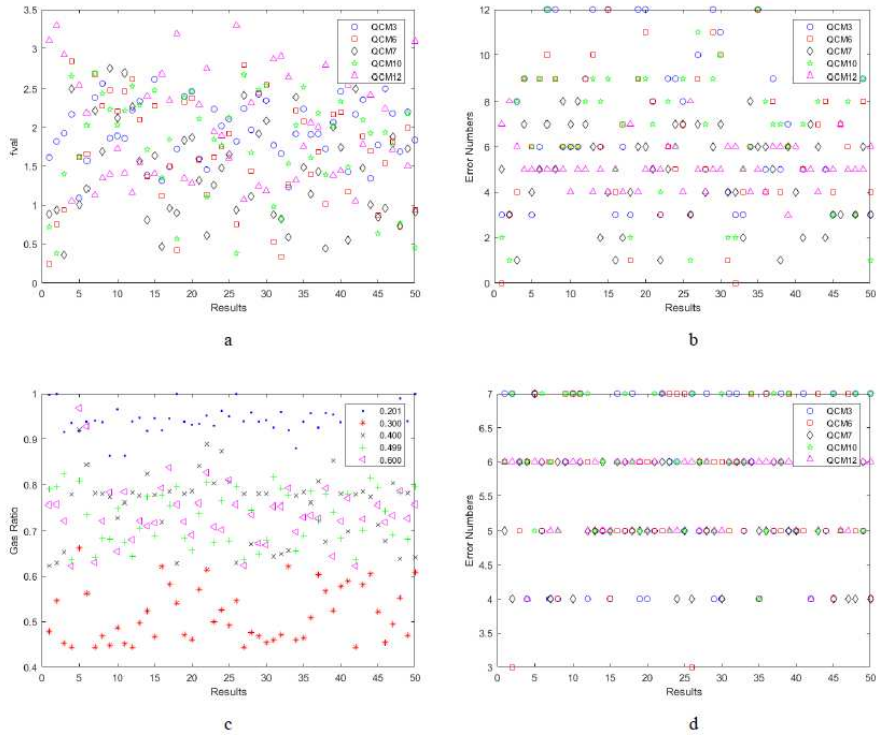

Figure 4

The results obtained for the scenario where each variable of the proposed method has 1 MF

Looking at graph “a” in Figure 4, it is seen that QCM6 and QCM7 can make more successful classification than with other sensors for the MOEFCs designed with the obtained solutions. Graph “b” shows that with 2 solutions in the set of Pareto optimal solutions, correct classification can be made with QCM6 in the model to be designed at all gas concentrations. The “c” graph shows that the environment

with a gas concentration of 0.201 is more effective on the solutions because the weights assigned for measurements made in this environment are at a higher level. The "d" graph shows that QCM6 can also produce successful solutions for test samples, and with the proposed method, error-free classification can be made in 2 design samples.

3.3.2 Scenarios That Have 2 MFs in Each Variable

Figure 5 shows the results obtained at the end of training and testing for MOEFC with 2 MFs in each variable.

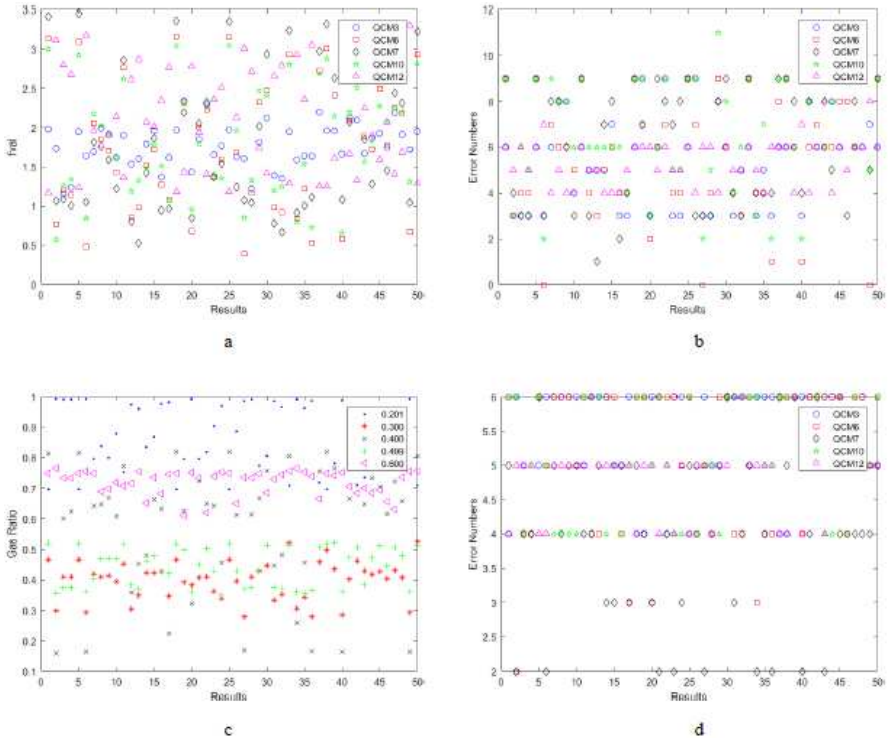

Figure 5
The results obtained for the scenario where each variable of the proposed method has 2 MFs

Figure 5 graphs show that the optimal solutions in the Pareto set are generally successful in favor of QCM6 and QCM7. The error levels obtained for these sensors and the classification errors are lower than other sensors' results. For the training dataset, error-free classification can be made by QCM6 in 3 different designs. In addition, by QCM6 and QCM7, 1 classification error can be obtained in 1 and 2 different MOEFC designs, respectively. Regarding the test dataset, by QCM6 and QCM7, 1 classification error can be obtained in 1 and 9 different designs, respectively. When the weights are examined, the weights determined for

the 0.201 and 0.600 gas concentration inputs are increased significantly compared to the results obtained with 1 MF in many examples. In contrast, the weights determined for the 0.300 and 0.499 gas concentration inputs are decreased.

3.3.3 Scenarios That Have 3 MFs in Each Variable

Figure 6 presents the experimental results obtained for the MOEFC model with 3 MFs in each variable.

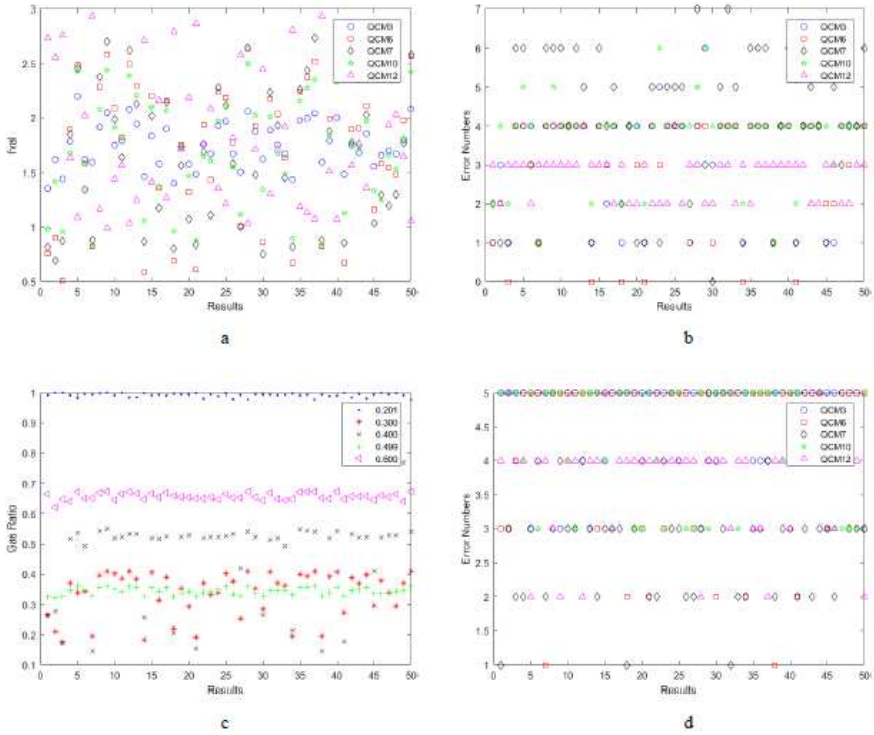

Figure 6

The results obtained for the scenario where each variable of the proposed method has 3 MFs

Figure 6 graphs show that the error rates obtained for the sensors are reduced to the 0.5-3.0 range compared to the results with 1 and 2 MF. In optimal, multi-objective solutions, better results are obtained in favor of the QCM6 sensor. Note that QCM12 errors are significantly higher, while the errors obtained with other sensors are low. On the other hand, in classifications made with other sensors, misclassifications are higher than QCM12. This contrast can be interpreted as the *μOut* values obtained with the QCM12 do not approach the cluster centers. In 6 different MOEFC designs, all the training data can be classified correctly by the QCM6. Also, all samples can be classified correctly by QCM7 in 1 design. However, in test samples, QCM7 is more successful. Error-free classification can

be made in 3 different designs by QCM7 and 2 by QCM6. When the weights are examined, it is seen that the weights of 0.201 gas concentration are superior to the others. An interesting result is that the weights cluster at specific intervals.

3.3.4 Scenarios That Have 4 MFs in Each Variable

Final experiments within the scope of the study are for MOEFCs with 4 MFs in each variable. In these experiments, it is expected that the accuracy is increases compared to the previous ones, but the interpretation ability of the model is expected to decrease. The obtained results are presented in Figure 7.

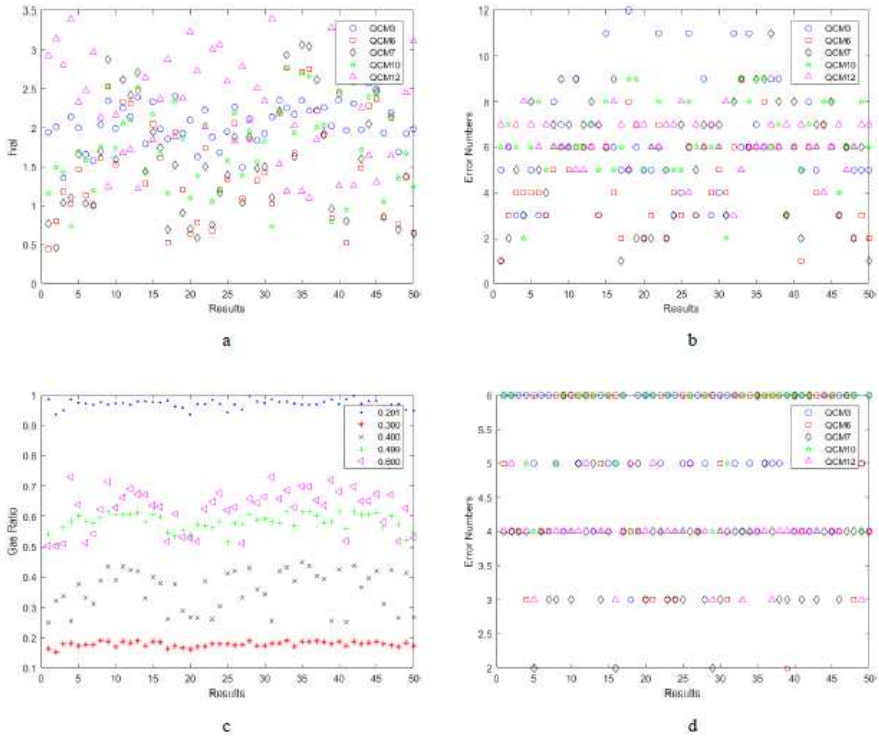

Figure 7
The results obtained for the scenario where each variable of the proposed method has 4 MFs

From Figure 7, it is seen that optimal, multi-objective solutions focus on the QCM6 and QCM7 sensors, like previous experiments. Most solutions that achieve low error levels achieve minimal error rates by these two sensors. The results are not different in terms of classification errors. However, although in many experiments, lower error rates are obtained by QCM3 compared to QCM12, the number of classification errors obtained with QCM3 is higher. In the classification made for the test dataset, the best success is achieved with 4 different designs that make 2 misclassifications. In 3 of these, the best classification can be made by

QCM7 and in one by QCM6. Weights are similarly in favor of 0.201 gas concentration. At a gas concentration of 0.300, the minimum weight coefficients are obtained.

3.4 Comparisons

In this section, the classification results of the proposed method for different sensors are evaluated. The classification results of the method with different parameter sets are examined, and its performance is compared with other MOEFC methods in the literature. Selected comparison algorithms are the Multi-Objective Differential Evolution Algorithm-based Fuzzy Clustering (MODEFC) [32], the NSGA-II-based Fuzzy Clustering (MOGAFC) [33], and Multi-Objective Modified Differential Evolution based Fuzzy Clustering (MOMoDEFC) [34] methods.

For each scenario, the solutions with the smallest total error (E_{SUM}) obtained for the objectives among the 50 optimal solutions in the Pareto solution set are given in Table 5 ($E_{SUM} = E_{QCM3} + E_{QCM6} + E_{QCM7} + E_{QCM10} + E_{QCM12}$).

Table 5

Solutions with the smallest total error obtained for each scenario

		E1	E2	E3	0.201	0.300	0.400	0.499	0.600
1 MF	QCM3	1.6652	3	5	0.9598	0.4710	0.6490	0.7905	0.7537
	QCM6	0.3350	0	5					
	QCM7	0.8267	4	6					
	QCM10	0.8272	2	6					
	QCM12	2.8977	6	6					
2 MFs	QCM3	1.4323	3	6	0.9918	0.3812	0.3207	0.3612	0.7520
	QCM6	0.6795	0	2					
	QCM7	0.8464	3	2					
	QCM10	0.9572	2	6					
	QCM12	2.7784	6	5					
3 MFs	QCM3	1.4411	1	2	0.9993	0.1759	0.1710	0.3274	0.6475
	QCM6	0.5080	0	1					
	QCM7	0.8708	1	2					
	QCM10	0.9562	2	2					
	QCM12	2.7622	3	3					
4 MFs	QCM3	1.6263	2	2	0.9701	0.1694	0.2657	0.5740	0.5175
	QCM6	0.7825	3	3					
	QCM7	0.5850	2	2					
	QCM10	1.3804	3	4					
	QCM12	2.7257	5	3					

The columns in Table 5 contain the following information:

E1: The error levels of the sensors, at the best classification, obtained by Eq. (11) for 50 samples

E2: Misclassification numbers of sensors for the training dataset for 50 samples, at the best classification

E3: Misclassification numbers of sensors for test dataset for 50 samples, at best classification

Columns 0.201, 0.300, 0.400, 0.499 and 0.600 show the error levels of the sensors obtained by Eq. (11) at these gas ratios.

When the results given in Table 5 are examined, it is seen that the classification success of the design with 3 MFs in each variable is higher than the other designs. Therefore, the algorithms model MOEFC with 3 MF in each variable in the comparison. For a fair comparison, the parameter settings for all algorithms have been assigned as in Table 4. The MCN is 10000 in each trial, and the algorithm has been run three times independently.

Table 6
Air-gas concentrations in experiments

		E1	**E2**	**E3**	**0.201**	**0.300**	**0.400**	**0.499**	**0.600**
MODEFC	QCM3	16.582	1	4	0.9375	0.3001	0.2788	0.3912	0.6963
	QCM6	0.531	1	4					
	QCM7	0.898	2	4					
	QCM10	0.817	3	5					
	QCM12	26.421	3	6					
MOGAFC	QCM3	14.715	2	2	0.9807	0.2189	0.2091	0.3964	0.7007
	QCM6	0.662	0	1					
	QCM7	0.883	2	3					
	QCM10	0.871	3	3					
	QCM12	25.458	3	3					
MOMoDEFC	QCM3	15.090	1	2	0.9468	0.2013	0.199	0.3117	0.6817
	QCM6	0.554	1	1					
	QCM7	0.937	2	3					
	QCM10	0.859	1	2					
	QCM12	28.796	4	5					
iwMOEFC	QCM3	14.411	1	2	0.9993	0.1759	0.171	0.3274	0.6475
	QCM6	0.508	0	1					
	QCM7	0.871	1	2					
	QCM10	0.956	2	2					
	QCM12	27.622	3	3					

The best results of the proposed method and other MOEFC methods are compared in Table 6. In Table 6, the proposed method is shortly named "*iwMOEFC*". While MODEFC and MOGAFC make a total of 10 misclassifications for the samples in the training dataset, MOMoDEFC makes 9, and iwMOEFC makes 7. However, the misclassification numbers of the algorithms for the test set are as follows: MODEFC=23, MOGAFC=12, MOMoDEFC=13 and iwMOEFC=10. It can also be seen from column E1 that iwMOEFC can classify values closer to the classification centers. In the E1 column, the distances of the classification values to the class centers are given. Accordingly, iwMOEFC has the minimum classification errors for the QCM3, QCM6 and QCM7 sensors. Moreover, iwMOEFC has produced minor error levels than other MOEFC methods at different gas ratios.

It has been observed that the solutions in the Pareto optimal set are generally successful in favor of the QCM6 sensor. In terms of weights, it is seen that more successful classifications can be made in an environment with a gas concentration of 0.201, and this environment is more effective in the general classification. However, measurements in an environment with a gas concentration of 0.300 have the lowest effect on classification.

Conclusions

The balance of accuracy and interpretability, one of the fundamental criteria in fuzzy system design, is particularly influential in system design and performance. EAs can successfully scan the problem's solution space by focusing on efficient solution regions in numerical optimization problems. These algorithms also provide adaptive training in many multi-objective fuzzy classifier methods, as they are not trapped in local optimal solutions. In this study, multi-objective EA is used for MOEFC design, but instead of a rule table, the weighted-input approach is applied for input-output interaction on the fuzzy logic side of the system. In this way, it can obtain information about the relative importance of the inputs for each objective in the problem.

The proposed method was used for alcohol classification. Alcohols were classified by evaluating the results obtained with different gas sensors in environments with different gas-air densities.

When the classification results are examined, it is proven that the proposed method can make successful classifications for many sensors simultaneously, with negligible error levels, even in environments with different gas-air densities. In addition, compared to other MOEFC methods, the performance of the method is more effective and provides superior solutions.

References

[1] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, “A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions,” *IEEE Trans. Fuzzy Syst.*, Vol. 21,

No. 1, pp. 45-65, Feb. 2013

[2] F. Jiménez, G. Sánchez, and J. M. Juárez, "Multi-objective evolutionary algorithms for fuzzy classification in survival prediction," *Artif. Intell. Med.*, Vol. 60, No. 3, pp. 197-219, Mar. 2014

[3] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *Int. J. Approx. Reason.*, Vol. 44, No. 1, pp. 4-31, Jan. 2007

[4] H. Ishibuchi, T. Murata, and I. Türkşen, "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets Syst.*, Vol. 89, No. 2, pp. 135-150, Jul. 1997

[5] T. Suzuki, T. Furuhashi, S. Matsushita, and H. Tsutsui, "Efficient fuzzy modeling under multiple criteria by using genetic algorithm," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, 1999, Vol. 5, pp. 314-319

[6] M. B. Gorzalczany and F. Rudzinski, "A multi-objective-genetic-optimization-based data-driven fuzzy classifier for technical applications," in *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, 2016, Vol. 2016-Novem, No. 3, pp. 78-83

[7] H. Ishibuchi, Y. Nojima, and I. Kuwajima, "Fuzzy Data Mining by Heuristic Rule Extraction and Multiobjective Genetic Rule Selection," in *2006 IEEE International Conference on Fuzzy Systems*, 2006, pp. 1633-1640

[8] P. Ducange, G. Mannara, F. Marcelloni, R. Pecori, and M. Vecchio, "A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classifiers," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1-6

[9] P. Pulkkinen, J. Hytönen, and H. Koivisto, "Developing a bioaerosol detector using hybrid genetic fuzzy systems," *Eng. Appl. Artif. Intell.*, Vol. 21, No. 8, pp. 1330-1346, Dec. 2008

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, Vol. 6, No. 2, pp. 182-197, Apr. 2002

[11] C. Setzkorn, A. F. G. Taktak, and B. E. Damato, "On the use of multi-objective evolutionary algorithms for survival analysis," *Biosystems*, Vol. 87, No. 1, pp. 31-48, Jan. 2007

[12] A. G. Di Nuovo and V. Catania, "An Efficient Approach for the Design of Transparent Fuzzy Rule-Based Classifiers," in *2006 IEEE International Conference on Fuzzy Systems*, 2006, pp. 1381-1387

[13] M.-S. Kim, C.-H. Kim, and J.-J. Lee, "Evolving Compact and Interpretable Takagi–Sugeno Fuzzy Models With a New Encoding Scheme," *IEEE Trans. Syst. Man Cybern. Part B*, Vol. 36, No. 5, pp. 1006-1023, Oct. 2006

[14] R. Vaishali, R. Sasikala, S. Ramasubbareddy, S. Remya, and S. Nalluri, "Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 2017, Vol. 2017-Janua, pp. 1-5

[15] Y. Nojima and H. Ishibuchi, "Designing Fuzzy Ensemble Classifiers by Evolutionary Multiobjective Optimization with an Entropy-Based Diversity Criterion," in *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06)*, 2006, No. 1, pp. 59-59

[16] C. A. Coello Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Comput. Intell. Mag.*, Vol. 1, No. 1, pp. 28-36, Feb. 2006

[17] A. Bal and S. I. Satoglu, "The use of Data Envelopment Analysis in evaluating Pareto optimal solutions of the sustainable supply chain models," *Procedia Manuf.*, Vol. 33, pp. 485-492, 2019

[18] A. Mukhopadhyay and U. Maulik, "Unsupervised Pixel Classification in Satellite Imagery Using Multiobjective Fuzzy Clustering Combined With SVM Classifier," *IEEE Trans. Geosci. Remote Sens.*, Vol. 47, No. 4, pp. 1132-1138, Apr. 2009

[19] A. Rodríguez-Molina, E. Mezura-Montes, M. G. Villarreal-Cervantes, and M. Aldape-Pérez, "Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem," *Appl. Soft Comput.*, Vol. 93, p. 106342, Aug. 2020

[20] N. Srinivas and K. Deb, "Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evol. Comput.*, Vol. 2, No. 3, pp. 221-248, Sep. 1994

[21] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evol. Comput.*, Vol. 8, No. 2, pp. 173-195, Jun. 2000

[22] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Overview of NSGA-II for Optimizing Machining Process Parameters," *Procedia Eng.*, Vol. 15, pp. 3978-3983, 2011

[23] Matlab, *Fuzzy Logic Toolbox ™ User's Guide R 2021a*, 1995th-2021st ed. The MathWorks, Inc., 2014

[24] H. Zheng, B. Jiang, and H. Lu, "An adaptive neural-fuzzy inference system (ANFIS) for detection of bruises on Chinese bayberry (Myrica rubra) based on fractal dimension and RGB intensity color," *J. Food Eng.*, Vol. 104, No.

4, pp. 663-667, Jun. 2011

[25] A. Marjani, M. Babanezhad, and S. Shirazian, "Application of adaptive network-based fuzzy inference system (ANFIS) in the numerical investigation of Cu/water nanofluid convective flow," *Case Stud. Therm. Eng.*, Vol. 22, No. November, p. 100793, Dec. 2020

[26] R. M. C. Ratnayake and K. Antosz, "Development of a Risk Matrix and Extending the Risk-based Maintenance Analysis with Fuzzy Logic," *Procedia Eng.*, Vol. 182, No. 1877, pp. 602-610, 2017

[27] M. F. Adak, P. Lieberzeit, P. Jarujamrus, and N. Yumusak, "Classification of alcohols obtained by QCM sensors with different characteristics using ABC based neural network," *Eng. Sci. Technol. an Int. J.*, Vol. 23, No. 3, pp. 463-469, Jun. 2020

[28] F. W. Wibowo and W. Wihayati, "Multi-classification of Alcohols using Quartz Crystal Microbalance Sensors based-on Artificial Neural Network Single Layer Perceptron," in *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*, 2021, pp. 37-41

[29] B. MILLER and J. KRIM, *Encyclopedia of Tribology*. Boston, MA: Springer US, 2013

[30] N. N. Htun, "Classification of Alcohols in Cosmetic Production," *Annu. Univ. J. Res. Appl.*, Vol. 1, No. 1, pp. 1-5, 2019

[31] L. A. Zadeh, A. M. Abbasov, and S. N. Shahbazova, "Fuzzy-Based Techniques in Human-Like Processing of Social Network Data," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, Vol. 23, No. Suppl. 1, pp. 1-14, Dec. 2015

[32] I. Saha, "Use of multiobjective differential fuzzy clustering with ANN classifier for unsupervised pattern classification: Application to microarray analysis," in *2010 Fourth International Conference on Research Challenges in Information Science (RCIS)*, 2010, pp. 647-656

[33] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective Genetic Clustering for Pixel Classification in Remote Sensing Imagery," *IEEE Trans. Geosci. Remote Sens.*, Vol. 45, No. 5, pp. 1506-1511, May 2007

[34] I. Saha, U. Maulik, and D. Plewczynski, "A new multi-objective technique for differential fuzzy clustering," *Appl. Soft Comput.*, Vol. 11, No. 2, pp. 2765-2776, Mar. 2011

Fuzzy Methods for Comparing Project Situations and Selecting Precedent Decisions

Vadim Borisov[1], Margarita Chernovalova[1], Marina Dulyasova[2], Dmitry Morozov[3], Artem Vasiliev[4]

[1]Branch of the National Research University Moscow Power Engineering Institute in Smolensk, 1, Energeticheskiy Proezd, Smolensk, 214013, Russia, borisovvadv@mpei.ru, ChernovalovaMV@mpei.ru

[2]Pushchino State Institute of Natural Sciences, 3, Prospekt Nauki, Pushchino, 142290, Russia, rector@pushgu.ru

[3]BIOCAD, 34-A, Svyazi st., Strelna, Saint-Petersburg, 198515, Russia, morozov@biocad.ru

[4]Moscow University for Industry and Finance «Synergy», 80, Leningradsky Prospect, Moscow, 125190, Russia, synergy@synergy.ru

Abstract: This paper characterizes project management task aspects substantiating the expediency of applying fuzzy methods for comparing project situations and selecting precedent decisions. It discusses methods for assessing the similarity of the fuzzy features of project situations, based on operations with fuzzy sets, pseudometric distances between fuzzy sets, and the fuzzy distance between fuzzy sets. The paper also describes approaches to comparing fuzzy project situations on the basis of aggregating the results of comparing individual features with the use of various convolutions or fuzzy inference algorithms, as well as by individual priority features. An example of selecting precedent project decisions relevant to project situations is given, where relative pseudometric distance between fuzzy sets is used to estimate the degree of similarity among the fuzzy features of project situations, and the modifiable Mamdani fuzzy inference algorithm is used for comparing fuzzy project situations and selecting precedent project decisions.

Keywords: project management; fuzzy project situations; fuzzy distance; fuzzy logic inference; precedent decision

1 Introduction

Case-based reasoning methods are currently used for making effective project decisions [1-3]. The characteristic features of project management tasks, which substantiate the advisability of applying these methods, are as follows:

- the "non-stationarity" of the conceptual and terminological tools, the rapidly changing structure and parameters of the project management subject area [4, 5];
- the incompleteness and insufficiency of information on the project situations to be compared, including the expert nature of information on project situation features and their heuristic representation [6];
- temporal and resource limitations imposed on the formation and selection of precedent decisions;
- the complexity of the homogeneous representation of project situations and precedent project decisions;
- different "scale" of project situations, herewith implying elaboration of similar project decisions [7];
- the complexity of estimating the similarity of the fuzzy features of project situations;
- the complexity of comparing project situations due to different compositions of the features, their different significances and degree of consistency;
- the task of selecting precedent project decisions relevant to project situations generally reduces to the task of classification and depends on the corresponding method of comparing fuzzy project situations [8].

The above mentioned causes a contradiction between the necessity to increase the degree of the reasonableness of project management decisions by means of the application of automated procedures of data accumulation and processing and a certain imperfection of the currently available decision support methods in terms of taking into account the specific features of innovative projects.

The characteristic features of project management tasks mentioned in the Introduction justify the expediency of using the representation of the features of project situations and project precedent decisions in the form of fuzzy sets and fuzzy relations.

We introduce the following nomenclature:

$Q_l = \left\{ \tilde{q}_n^{(l)} \mid n = 1, ..., N \right\}$ is the l-th typical fuzzy project situation (l = 1,…, L) represented by fuzzy sets (numbers) $\tilde{q}_n^{(l)}$ of its features;

$P_k = \left\{ \tilde{p}_n^{(k)} \mid n = 1, ..., N \right\}$ is the k-th current project situation (k = 1,…, K) represented by fuzzy sets (numbers) $\tilde{p}_n^{(k)}$ of its features; N is the number of features to be compared.

This way it is possible to perform fuzzy granulation and to determine the relevance between a project situation and a precedent decision.

The paper presents the analysis of fuzzy methods for comparing project situations for selecting relevant precedent decisions.

2 Assessment of Similarity between the Corresponding Features of Project Situations to be Compared

The method for assessing the degree of similarity between the respective features of the project situations to be compared must meet the following requirements:

1) The method must not only indicate which feature is more/less significant, but also enable one to judge about the significance of the difference in the features.

2) The method must preserve the adequacy of similarity assessment: for fuzzy values of features with disjoint supports; for coinciding fuzzy values of features; for crisp values of features.

3) The method must take into account the form of membership functions for fuzzy values of features with disjoint supports. Herewith, it is desirable that high values of the α-levels of features being compared have a greater effect on the comparison result.

4) The method must enable one to compare fuzzy values of features different in both the width of the basic range and the form of the membership functions.

Based on the formulated requirements, to assess the similarity between the corresponding features of project situations to be compared, methods based on the following approaches can be applied:

- operations with fuzzy sets (disjunctive sum, bounded difference, disjoint sum, etc.) [9-14];
- pseudometric distance between fuzzy sets (Hamming distance, Euclidean distance, etc.) [15-20];
- ranking indexes for fuzzy sets (numbers) [21];
- logical indexes for comparing fuzzy sets (numbers) [22].

2.1 Application of Operations with Fuzzy Sets

Fuzzy set operations defined in general form through *t*-norms and *s*-norms can be used for assessing the similarity of fuzzy values of features. In what follows, we offer the most widespread examples of such operations.

1) The difference $\left(\tilde{q}_n^{(l)} - \tilde{p}_n^{(k)}\right)$ with the membership function

$$\mu_{\left(\tilde{q}_n^{(l)} - \tilde{p}_n^{(k)}\right)}(x) = \min\left(\mu_{\tilde{q}_n^{(l)}}(x), \left(1 - \mu_{\tilde{p}_n^{(k)}}(x)\right)\right), \ \forall x \in X, \tag{1}$$

where X is a universal set on which the fuzzy sets $\tilde{q}_n^{(l)}$ and $\tilde{p}_n^{(k)}$ are specified.

2) The bounded difference $\left(\tilde{q}_n^{(l)} \theta \tilde{p}_n^{(k)}\right)$ with the membership function

$$\mu_{\left(\tilde{q}_n^{(l)} \theta \tilde{p}_n^{(k)}\right)}(x) = \max\left(0, \mu_{\tilde{q}_n^{(l)}}(x) - \mu_{\tilde{p}_n^{(k)}}(x)\right), \ \forall x \in X. \tag{2}$$

3) The disjunctive sum $\left(\tilde{q}_n^{(l)} \oplus \tilde{p}_n^{(k)}\right)$ with the membership function

$$\mu_{\left(\tilde{q}_n^{(l)} \oplus \tilde{p}_n^{(k)}\right)}(x) = \max\left(\min\left(\mu_{\tilde{q}_n^{(l)}}(x), \left(1 - \mu_{\tilde{p}_n^{(k)}}(x)\right)\right), \min\left(\left(1 - \mu_{\tilde{q}_n^{(l)}}(x)\right), \mu_{\tilde{p}_n^{(k)}}(x)\right)\right), \tag{3}$$

$\forall x \in X$.

4) The disjoint sum $\left(\tilde{q}_n^{(l)} \Delta\, \tilde{p}_n^{(k)}\right)$ with a membership function:

$$\mu_{\left(\tilde{q}_n^{(l)} \Delta\, \tilde{p}_n^{(k)}\right)}(x) = \left|\mu_{\tilde{q}_n^{(l)}}(x) - \mu_{\tilde{p}_n^{(k)}}(x)\right|, \ \forall x \in X\,. \tag{4}$$

The choice of this or that operation with fuzzy sets leads to different results of assessing the similarity of fuzzy features. Such a choice is justified by the identified conditions of comparing project situations and by the system of preferences of the decision making person.

Example. For $\tilde{q}_n^{(l)} = \{0.1/x_1, 0.5/x_2, 1.0/x_3, 0.7/x_4, 0.3/x_5\}$ and $\tilde{p}_n^{(k)} = \{0.2/x_1, 0.4/x_2, 0.6/x_3, 0.8/x_4, 1.0/x_5\}$, the operation results are as follows:

$\left(\tilde{q}_n^{(l)} - \tilde{p}_n^{(k)}\right) = \{0.1/x_1, 0.5/x_2, 0.4/x_3, 0.2/x_4, 0.0/x_5\}$; $\left(\tilde{q}_n^{(l)} \theta \tilde{p}_n^{(k)}\right) = \{0.0/x_1, 0.1/x_2, 0.4/x_3, 0.0/x_4, 0.0/x_5\}$; $\left(\tilde{q}_n^{(l)} \oplus \tilde{p}_n^{(k)}\right) = \{0.2/x_1, 0.5/x_2, 0.4/x_3, 0.3/x_4, 0.7/x_5\}$; $\left(\tilde{q}_n^{(l)} \Delta\, \tilde{p}_n^{(k)}\right) = \{0.1x_1/, 0.1/x_2, 0.4/x_3, 0.1/x_4, 0.7/x_5\}$.

2.2 Application of Pseudometric Distances between Fuzzy Sets

The main types of pseudometric distances for assessing the degree of similarity of analogous features of project situations, represented by the fuzzy sets $\tilde{q}_n^{(l)}$ and $\tilde{p}_n^{(k)}$, are the Hamming and Euclidean distances between fuzzy sets [23, 24].

The relative Hamming distance between fuzzy sets is as follows:

$$d_H\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \frac{1}{n}\sum_{i=1}^{n}\left|\mu_{\tilde{q}_n^{(l)}}(x_i) - \mu_{\tilde{p}_n^{(k)}}(x_i)\right|, \ x_i \in X. \tag{5}$$

The relative Euclidean distance between fuzzy sets is

$$d_E\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^{n} \left(\mu_{\tilde{q}_n^{(l)}}(x_i) - \mu_{\tilde{p}_n^{(k)}}(x_i)\right)^2}, \; x_i \in X. \tag{6}$$

Example. For $\tilde{q}_n^{(l)}$ = {0.3/x_1, 1.0/x_2, 0.4/x_3, 0.0/x_4} and $\tilde{p}_n^{(k)}$ = {0.2/x_1, 0.5/x_2, 1.0/x_3, 0.0/x_4}: $d_H\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = 0.3$ and $d_E\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = 0.395$.

The assessment resulting from the application of pseudometric distances does not require any defuzzification. On the one hand, this facilitates subsequent aggregation of the results of feature-by-feature comparison (as distinct from the application of the previously described operations with fuzzy sets); on the other hand, this is characterized by a lower possibility of taking into account conditions for determining the relevance of the project situations to be compared and the system of preferences of the decision making person.

The pseudometric distances discussed above are treated conventionally, while the application of L. A. Zadeh's generalization principle enables us to treat distance as a fuzzy set as follows [25, 26]:

$$\forall \delta \in \Re^+ \qquad \tilde{d}\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \max_{\delta \in \tilde{d}\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)} \left(\min\left(\mu_{\tilde{q}_n^{(l)}}(x_i), \mu_{\tilde{p}_n^{(k)}}(x_i)\right)\right), \; \forall x_i \in X. \tag{7}$$

where $\Re^+$ – the set of non-negative numbers.

Example. For $\tilde{q}_n^{(l)}$ = {0.1/x_1, 0.5/x_2, 1.0/x_3, 0.7/x_4, 0.3/x_5} and $\tilde{p}_n^{(k)}$ = {0.2/b_1, 0.4/b_2, 0.6/b_3, 0.8/b_4, 1.0/b_5}: $\tilde{d}\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)$ = {0.7/δ_1, 0.8/δ_2, 1.0/δ_3, 0.5/δ_4, 0.2/δ_5}.

2.3 Application of Ranking Indexes for Fuzzy Sets (Numbers)

In this section we give examples of the most widespread indexes for ranking fuzzy sets (numbers).

1) The fuzzy set (number) ranking index based on the fuzzy preference relation:

$$I_1\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \sup_{(x_1, x_2) \in Supp\left(\tilde{q}_n^{(l)}\right) \times Supp\left(\tilde{p}_n^{(k)}\right)} \min\left(\mu_{\tilde{q}_n^{(l)}}(x_1), \mu_{\tilde{p}_n^{(k)}}(x_2), \mu_Q(x_1, x_2)\right). \tag{8}$$

This index uses the fuzzy preference relation Q on $\Re^2$, e.g. with the membership function

$$\mu_Q(x_1, x_2) = \begin{cases} 1, x_1 \geq x_2, \\ 0, x_1 < x_2. \end{cases} \tag{9}$$

The ranking of $\tilde{q}_n^{(l)}$ and $\tilde{p}_n^{(k)}$ is performed according to the rule

$$I_1\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) \geq I_1\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) \Rightarrow \tilde{q}_n^{(l)} \geq \tilde{p}_n^{(k)}. \quad (10)$$

2) The fuzzy set (number) ranking index based on comparing their mean values:

$$I_2\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) \geq m\left(\tilde{q}_n^{(l)}\right) - m\left(\tilde{p}_n^{(k)}\right), \quad (11)$$

where $m\left(\tilde{q}_n^{(l)}\right), m\left(\tilde{p}_n^{(k)}\right)$ is the mean values of the fuzzy numbers $\tilde{q}_n^{(l)}$ and $\tilde{p}_n^{(k)}$, respectively.

The sign and value of the index $I_2\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)$ are indicative of what fuzzy number is greater and how much.

3) The index of ranking the fuzzy numbers $\tilde{q}_n^{(l)}$ and $\tilde{p}_n^{(k)}$, based on the membership function of the fuzzy number $\tilde{D} = \dfrac{\tilde{q}_n^{(l)}}{\tilde{q}_n^{(l)} + \tilde{p}_n^{(k)}}$ [22]:

$$I_3\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \int_0^{0.5} (1 - \mu_{\tilde{D}}(z))dz + \int_{0.5}^{1} \mu_{\tilde{D}}(z)dz, \quad (12)$$

$$\mu_{\tilde{D}}(z) = \sup_{z = x_1/(x_1+x_2)} \min(\mu_{\tilde{q}_n^{(l)}}(x_1), \mu_{\tilde{p}_n^{(k)}}(x_2)). \quad (13)$$

$$I_3\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) \geq 0.5 \Rightarrow A \geq B. \quad (14)$$

4) The fuzzy number ranking indexes proposed by D. Dubois and H. Prade, and based on seeking the highest/lowest value of the membership function among pairs of elements of fuzzy number supports [21]:

$$I_4^1\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \sup_{\substack{(x_1, x_2) \in Supp\left(\tilde{q}_n^{(l)}\right) \times Supp\left(\tilde{p}_n^{(k)}\right) \\ x_1 \geq x_2}} \min\left(\mu_{\tilde{q}_n^{(l)}}(x_1), \mu_{\tilde{p}_n^{(k)}}(x_2)\right), \quad (15)$$

$$I_4^2\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \sup_{x_1 \in Supp\left(\tilde{q}_n^{(l)}\right)} \inf_{\substack{x_2 \in Supp\left(\tilde{p}_n^{(k)}\right) \\ x_2 \geq x_1}} \min\left(\mu_{\tilde{q}_n^{(l)}}(x_1), 1 - \mu_{\tilde{p}_n^{(k)}}(x_2)\right), \quad (16)$$

$$I_4^3\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \inf_{x_1 \in Supp\left(\tilde{q}_n^{(l)}\right)} \sup_{\substack{x_2 \in Supp\left(\tilde{p}_n^{(k)}\right) \\ x_2 \leq x_1}} \max\left(1 - \mu_{\tilde{q}_n^{(l)}}(x_1), \mu_{\tilde{p}_n^{(k)}}(x_2)\right), \quad (17)$$

$$I_4^4\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = 1 - \sup_{\substack{(x_1, x_2) \in Supp\left(\tilde{q}_n^{(l)}\right) \times Supp\left(\tilde{p}_n^{(k)}\right) \\ x_1 \leq x_2}} \min\left(\mu_{\tilde{q}_n^{(l)}}(x_1), \mu_{\tilde{p}_n^{(k)}}(x_2)\right), \quad (18)$$

$$I_4^i\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) \geq I_4^i\left(\tilde{p}_n^{(k)}, \tilde{q}_n^{(l)}\right) \Rightarrow \tilde{q}_n^{(l)} \geq \tilde{p}_n^{(k)}, i \in \{1, ..., 4\}. \quad (19)$$

The ranking indexes $I_1\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)$, $I_4^1\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right), ..., I_4^4\left(\tilde{p}_n^{(k)}, \tilde{q}_n^{(l)}\right)$ ignore the form of the membership functions of $\tilde{q}_n^{(l)}$ and $\tilde{p}_n^{(k)}$.

The ranking index $I_2\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)$ takes into account the form of the membership functions, but its values are not normalized, and this complicates interpretation of assessment results.

The values of the ranking index $I_3\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)$ are normalized; however, it should be used for comparing non-negative fuzzy numbers or with allowance for the shift of the membership functions of fuzzy numbers being compared.

Note that the mean value $m\left(I_3\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)\right)$ of the index $I_3\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)$ has a clearer quantitative interpretation than the latter.

2.4 Application of Logical Indexes for Comparing Fuzzy Sets (Numbers)

The approach based on logical operations is applicable to assessing the similarity of the fuzzy values of project situation features. Herewith, the values of feature membership functions are treated as the truth degrees of a statement, and base set elements, in their turn, are taken into account in the determination of the truth/falsity of this statement. Therefore, the task is to determine the logical interrelation, i.e., whether the truth of the statement about the membership of the element in one fuzzy number entails the truth of a similar statement with respect to another fuzzy number.

The typical representation of logical indexes for comparing fuzzy numbers [22] is as follows:

$$ml\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \min_{x \in \Re} f\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right). \tag{20}$$

The following operations are most often used as $f\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right)$:

- fuzzy inclusion of the fuzzy number $\tilde{q}_n^{(l)}$ into the fuzzy number $\tilde{p}_n^{(k)}$, with the membership function

  $$\mu_f(x) = \mu_{\tilde{q}_n^{(l)}}(x) \to \mu_{\tilde{p}_n^{(k)}}(x), \forall x \in \Re, \tag{21}$$

  where $\to$ is fuzzy implication operation;

- fuzzy equality (equivalence) of the fuzzy numbers $\tilde{q}_n^{(l)}$ and $\tilde{p}_n^{(k)}$, with the membership function

$$\mu_f(x) = \left(\mu_{\tilde{q}_n^{(l)}}(x) \to \mu_{\tilde{p}_n^{(k)}}(x)\right) \mathrm{T} \left(\mu_{\tilde{p}_n^{(k)}}(x) \to \mu_{\tilde{q}_n^{(l)}}(x)\right), \forall x \in \Re, \tag{22}$$

where T $\to$ is t-norm operation, e.g., min.

The fuzzy inclusion operation is used when the falling of the fuzzy feature $\tilde{q}_n^{(l)}$ of the number into a class described by the reference fuzzy feature $\tilde{p}_n^{(k)}$ is sufficient. The fuzzy equality is typical for cases when it is required to determine the maximum coincidence of fuzzy features.

The result of comparing fuzzy numbers is much dependent on selecting the implementation of a fuzzy implication operation. Thus, the Larsen and Mamdani fuzzy implication operations do not suit the goals discussed since the result of the pointwise integration of fuzzy numbers is nonzero only if the supports of both fuzzy numbers coincide with the base set.

The Lukasiewicz, Gödel, Kleene–Dienes, and Kleene–Dienes–Lukasiewicz fuzzy implication operations yield equally correct results for the case of the full inclusion of the $\tilde{q}_n^{(l)}$ support into the set of modal $\tilde{p}_n^{(k)}$ values. However, in the case of complete equality between $\tilde{q}_n^{(l)}$ and $\tilde{q}_n^{(l)}$, the Kleene–Dienes and Kleene–Dienes–Lukasiewicz implication operations underestimate the degree of their compliance.

The logical index presented below is devoid of this limitation [27]:

$$ml\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right) = \max_{x \in \Re} \min f\left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)}\right). \tag{23}$$

3 Approaches to Comparing Fuzzy Project Situations

To compare fuzzy project situations for selecting relevant decisions, it is required to aggregate the results of comparing the fuzzy features of these situations. The aggregation of the results of comparing the fuzzy features of situations is generally based on one of the following approaches:

- reduction of the multicriterion assessment task to the one-criterion one based on the aggregation of the results of comparing individual features with the use of various convolutions (additive, multiplicative, maximin, minimax, etc.) or fuzzy inference algorithms (by Mamdani, Larsen, Takagi-Sugeno, Tsukamoto, etc.);
- by priority features, the other being considered as additional, whose comparison results must meet the established rules.

In the comparison of project situations, the former approach prevails, i.e., that based on the aggregation of the results of comparing individual features. Besides, sometimes the task of aggregating the results of comparing the features of project situations is solved "automatically". Problems arise in aggregation depending on various comparison conditions; namely, if

- different scales are used to compare different features;
- it is necessary to take into account different personal significances of the features;
- it is required do take into account the effect of consistency (including correlation and interplay) of the features on the overall result of comparing the situations;
- the project situations to be compared are characterized by a complex "structure" of feature aggregation.

Depending on these and some other conditions, the following strategies for aggregating the results of comparing the features of project situations are possible:

- the overall result of comparing project situations is represented as a hierarchy of partial results of feature comparison;
- the overall result of comparing project situations is formed under conditions of the equivalence of feature comparison results for the following instances:
  - "simultaneous" achievement of all the partial feature comparison results,
  - achievement of one of the partial feature comparison results,
  - compromise (intermediate) achievement of partial feature comparison results (e.g., achievement of individual partial results of feature comparison),
  - hybrid strategies targeted at the selection (identification) of convolution operations depending on the obtained values of partial feature comparison results [28];
- the overall result of comparing project situations is based on recursive aggregation of partial feature comparison results;
- the overall result of comparing project situations is formed under conditions of the inequivalence of partial feature comparison results for the following instances:
  - achievement of the required threshold values of partial feature comparison results,

- different weights for partial feature comparison results and taking them into account in subsequent aggregation, for example, using ordered weighted averaging (OWA) operators [29],
- a hierarchical AND/OR tree structure of feature aggregation.

- the overall result of comparing project situations is based on various quantifiers (including fuzzy) for the convolution of feature comparison results, e.g., in terms of the consistency of most of the features, in terms of the inconsistency of at least one feature.

4 An Example of Selecting Precedent Project Decisions Relevant to Fuzzy Project Situations

We use the relative Euclidean pseudometric distance between fuzzy sets to assess the similarity of the fuzzy features of project situations, and we use the modified Mamdani fuzzy inference algorithm for comparing fuzzy project situations and selecting precedent project decisions [30, 31].

In view of these conditions, the fuzzy model of comparing fuzzy project situations and selecting precedent project decisions can be represented as

$$\tilde{R}_{\Sigma} = \bigcup_{p=1,\ldots,P} \left(\min(\min_{n=1,\ldots,N} \left(d_E \left(\tilde{q}_n^{(l)}, \tilde{p}_n^{(k)} \right), \tilde{R}_p \right) \right), \tag{24}$$

where $\tilde{R}_{\Sigma}$ is an output fuzzy variable, whose value corresponds to the precedent project decision being selected; P is the number of fuzzy model rules.

Figure 1 illustrates the example of comparing fuzzy project situations and selecting precedent project decisions.

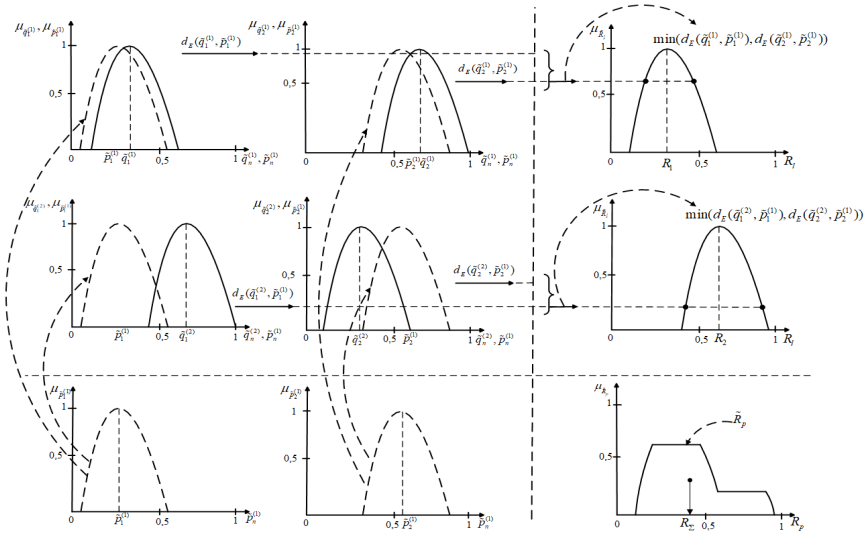

Figure 1
An example of comparing fuzzy project situations and selecting precedent project decisions

5 Software Implementation

To solve the task of comparing fuzzy project situations and selecting precedent project decisions, the *Loginf* program module has been developed in *Python*. The *matplotlib* library was used for result visualization.

Figure 2 shows the listing of the program of comparing fuzzy project situations and selecting precedent project decisions (for 2 features and 2 rules).

```
from matplotlib import pyplot as plt
import numpy as np
import loginf
base_set = np.arange(0, 1.01, 0.01) # Base Set
# RULE 1:
promise11 = loginf.FuzzyGaussian(base_set, 0, 0.14) # Promise 1
promise12 = loginf.FuzzyGaussian(base_set, 1, 0.14) # Promise 2
consequent1 = loginf.FuzzyGaussian(base_set, 0.6, 0.1) # Consequent 1
rule1 = loginf.FuzzyRule(consequent1, promise11, promise12) # Rule 1
# RULE 2:
promise21 = loginf.FuzzyGaussian(base_set, 1, 0.14) # Promise 1
promise22 = loginf.FuzzyGaussian(base_set, 0, 0.14) # Promise 2
consequent2 = loginf.FuzzyGaussian(base_set, 0.37, 0.1) # Consequent 2
rule2 = loginf.FuzzyRule(consequent2, promise21, promise22) # Rule 2
# TEST INPUT SETS:
# Input fuzzy sets:
input_set1 = loginf.FuzzyGaussian(base_set, 0.38, 0.05)
input_set2 = loginf.FuzzyGaussian(base_set, 0.68, 0.1)
```

```
# RULE OUTPUTS:
rule_output1 = rule1.consequent_output(input_set1, input_set2, plot = True)
rule_output2 = rule2.consequent_output(input_set1, input_set2, plot = True)
result = rule_output1.union(rule_output2) # Union output set
center = result.center_of_gravity() # Centroid
```

Figure 2

The program of comparing fuzzy project situations and selecting precedent project decisions

The visualization of solving the task of comparing fuzzy project situations and selecting a precedent project decision is exemplified in Figure 3.

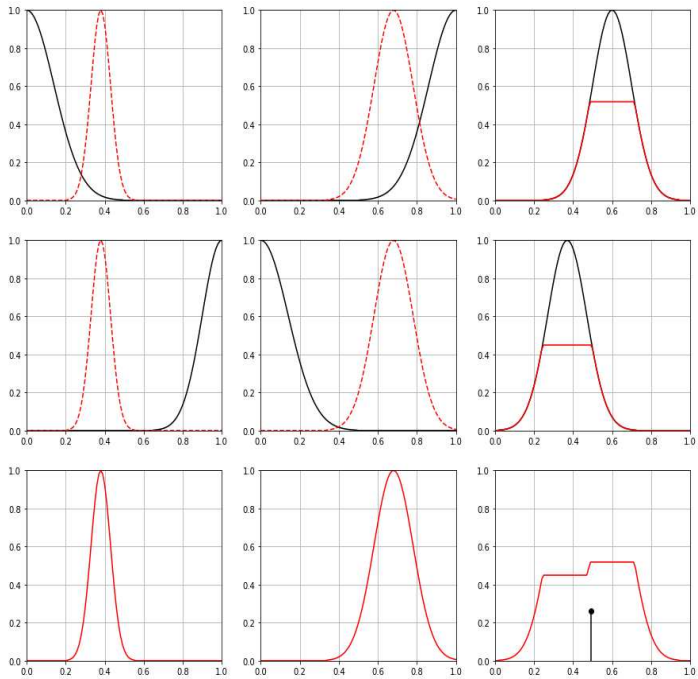

Figure 3

An example of using the *Loginf* program module

The main classes of this module are presented in Figure 4.

The following classes are used in the implementation of the *Loginf* module:

- GaussianFunction and FuzzyGaussian are used to specify fuzzy set membership functions;
- FuzzySet is intended for calculating pseudometric distances between the fuzzy features of precedent situations, performing operations with fuzzy sets (numbers), and defuzzifying the values of the fuzzy output variable;
- FuzzyRule implements the fuzzy inference algorithm.

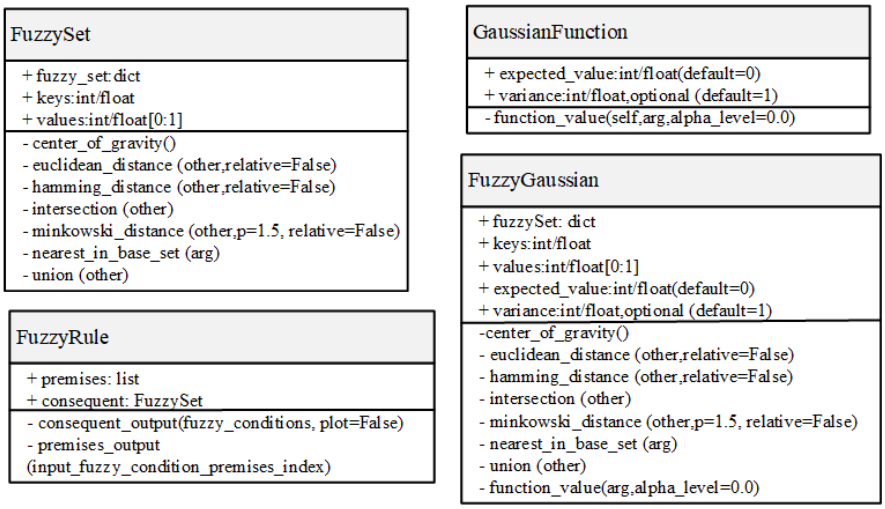

Figure 4
The main classes of the *Loginf* module

Conclusion

Methods for comparing fuzzy project situations have been analyzed and systematized.

Methods for assessing the similarity of the fuzzy features of project situations, based on operations with fuzzy sets, pseudometric distances between fuzzy sets, and the fuzzy distance between fuzzy sets have been discussed.

The paper has described approaches to comparing fuzzy project situations on the basis of transition from the multicriterion assessment task to the one-criterion one due to the aggregation of the results of comparing individual features with the use of various convolutions or fuzzy inference algorithms, as well as by individual priority features.

A program module has been developed and an example of selecting precedent project decisions relevant to project situations is given, where the relative pseudometric distance between fuzzy sets is used to assess the similarity of the fuzzy features of project situations, and the modified Mamdani fuzzy inference algorithm is used for comparing fuzzy project situations and selecting precedent project decisions.

Acknowledgement

This study was performed within the framework of the state assignment, project number FSWF-2023-0012.

References

[1] C. Tsatsoulis, A. B. Williams: Case-Based Reasoning, Knowledge-Based Systems, 2000, Vol. 3, pp. 807-837

[2] N. L. R. Machado, L. A. L. Silva, L. M. Fontoura, J. A Campbell: Case-based reasoning for experience-based collaborative risk management, Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, 2014, pp. 262-267

[3] M. J. Khan, H. Hayat, I. Awan: Hybrid case-based maintenance approach for modeling large scale case-based reasoning systems, Human-centric Computing and Information Sciences, 2019, Vol. 9, Iss.1, 9

[4] M. I. Dli, O. V. Bulygina, A. A. Emelyanov, Y. V Selyavskiy: Intelligent analysis of complex innovative project prospects, IOP Conference Series: Materials Science and Engineering, 2020, Vol. 919, Iss.4, 042019

[5] M. I. Dli, O. V. Stoyanova, I. V. Abramenkova, O. V Zaitsev: The method of intelligent management of information resources of an industrial enterprise, Journal of Applied Informatics, 2010, No. 5 (29), pp. 13-22

[6] O. Bulygina: Analysis of the feasibility of innovative projects for creating high technology products: the algorithms and instruments, Journal of Applied Informatics, 2016, Vol. 11, No. 4 (64), pp. 87-102

[7] M. Dli, A. Ofitserov, O. Stoianova, A. Fedulov: Complex model for project dynamics prediction, International Journal of Applied Engineering Research, 2016, Vol. 11, No. 22, pp. 11046-11049

[8] I. Chourib, G. Guillard, M. Mestiri, B. Solaiman, I. R. Farah: Case-Based Reasoning: Problems and Importance of Similarity Measur, Proceedings of the 2020 International Conference on Advanced Technologies for Signal and Image Processing, ATSIP, 2020, 9231755

[9] I. Batyrshin, O. Kosheleva, V. Kreinovich, N. Kubysheva, R. Akhtiamov: Contrast similarity measures of fuzzy sets, Computacion y Sistemas, 2019, Vol. 23. No. 4, pp. 1569-1573

[10] A. C. Aras, O. Kaynak, I. Batyrshin: Nonlinear function approximation based on fuzzy algorithms with parameterized conjunctors, Proceedings of the 2013 IEEE International Conference on Mechatronics, 2013, pp. 81-86, 6518515

[11] A. H. Zavala, I. Z. Batyrshin, O. C. Nieto, O. Castillo: Conjunction and disjunction operations for digital fuzzy hardware, Applied Soft Computing Journal, 2013, Vol. 13, No. 7, pp. 3248-3258

[12] I. Z. Batyrshin, I. J. Rudas, A. Panova: On generation of digital fuzzy parametric conjunctions, Studies in Computational Intelligence, 2009, Vol. 243, pp. 79-89

[13] I. Z. Batyrshin: Towards a general theory of similarity and association measures: Similarity, dissimilarity and correlation functions, J. Intell. Fuzzy Syst., 201, Vol. 36, No. 4, pp. 2977-3004

[14] A. H. Zavala, I. Z. Batyrshin, I. J. Rudas, L. V. Vargas, O. C. Nieto: Parametric operations for digital hardware implementation of fuzzy systems, Lecture Notes in Computer Science, 2009, Vol. 5845 LNAI, pp. 432-443

[15] W. S. Du: Subtraction and division operations on intuitionistic fuzzy sets derived from the Hamming distance, Information Sciences, 2021, Vol. 571, pp. 206-224

[16] Z. Md. Rodzi, N. A. Hassan: Hamming score distance of hesitant fuzzy sets, Journal of Physics: Conference Series, 2019, Vol. 1212, No. 1, 012020

[17] W. R. W. Mohd, L. Abdullah: Similarity measures of Pythagorean fuzzy sets based on combination of cosine similarity measure and Euclidean distance measure, AIP Conference Proceedings, 2018, Vol. 1974, 030017

[18] M.-S. Yang, Z. Hussain: Distance and similarity measures of hesitant fuzzy sets based on Hausdorff metric with applications to multi-criteria decision making and clustering, Soft Computing, 2019, Vol. 3, No. 14, pp. 5835-5848

[19] F. A. Xiao: Distance Measure for Intuitionistic Fuzzy Sets and Its Application to Pattern Classification Problems, Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics: Systems 2021, Vol. 51, No. 6, 8944285, pp. 3980-3992

[20] S. An, Q. Hu, C. Wang: Probability granular distance-based fuzzy rough set model, Applied Soft Computing, 2021, Vol. 102, 107064

[21] D. Dubois, H. Prade: A unifying view of comparison indices in a fuzzy set-theoretic framework, In: Fuzzy Sets and Possibility Theory – Recent Developments, R. R. Yager, Ed., Pergamon Press, 1982, pp. 1-13

[22] A. N. Borisov, A. V. Alekseev, G. V. Merkuryev: Obrabotka nechyotkoj informacii v sistemah prinyatiya reshenij [Processing of fuzzy information in decision-making systems]. Moscow, Radio and Communications, 1989, 304 p. (in Russia)

[23] M. Wygralak: On nonstrict archimedean triangular norms, hamming distances, and cardinalities of fuzzy sets, International Journal of Intelligent Systems, 2009, Vol. 24, No. 6, pp. 697-705

[24] R. Hidayat, I. T. R. Yanto, A. A. Ramli, M. F. M. Fudzee, A. S. Ahmar: Generalized normalized Euclidean distance based fuzzy soft set similarity for data classification, Computer Systems Science and Engineering, 2021, Vol. 38, No. 1, pp. 119-130

[25] L. A. Zadeh: Generalized Theory of Uncertainty (GTU)-Principal Concepts and Ideas, Computational Statistics and Data Analysis, 2006, Vol. 51, pp. 15-46

[26] V. V. Borisov, A. S. Fedulov, M. M. Zernov: Osnovy teorii nechetkikh mnozhestv [Fundamentals of fuzzy set theory]. Moscow, Goryachaya Liniya–Telekom Publ., 2014, 88 p. (in Russian)

[27] V. V. Borisov, A. S. Fedulov, M. M. Zernov: Osnovy nechetkoj arifmetiki [Fundamentals of fuzzy arithmetic]. Moscow, Goryachaya Liniya–Telekom Publ., 2014, 98 p. (in Russian)

[28] D. Dubois, H. Prade: A Review of Fuzzy Set Aggregation Connectives. Information Sciences, 1986, Vol. 39, pp. 105-210

[29] R. R. Yager, J. Kacprzyk: The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer: Norwell, MA, 1997

[30] M. V. Chernovalova: Fuzzy case models for project management using a multi-ontology approach, Journal of Applied Informatics, 2021, Vol. 16, No. 2, pp. 4-16

[31] V. V. Borisov, M. V. Chernovalova, S. P. Kurilin: Monitoring and adaptation of the base of design precedents in the management of innovative projects based on a fuzzy ontological approach, Ontology of Designing, 2020, Vol. 10, No. 4 (38), pp. 516-526

Comparison of the Three Algorithms for Concreteness Rating Estimation of English Words

Vladimir V. Bochkarev, Stanislav V. Khristoforov, Anna V. Shevlyakova, Valery D. Solovyev

Kazan Federal University, Kremlyovstaya 18, 420008 Kazan, Russia, vladimir.bochkarev@kpfu.ru, stanislav.khristoforov@tgtdiagnostics.com, AVShevlyakova@kpfu.ru, Valery.Solovyev@kpfu.ru

Abstract: The paper compares three algorithms for concreteness rating estimation of English words. To train and test the models, we used a number of freely available dictionaries containing concreteness ratings. A feedforward neural network is employed as a regression model. Pre–trained fastText vectors, data on co–occurrence of target words with the most frequent ones, and data on co–occurrence of target words with functional words are used as input data by the considered algorithms. One of the three algorithms was proposed for the first time in this article. We provide detailed explanations of which combinations with functional words are the most informative in terms of concreteness ratings estimation for English words. Although the rest two algorithms have already been used for estimation of concreteness ratings, we consider possible ways to update them and improve the results obtained by a neural network. Thuswise, we use stochastic Spearman's correlation coefficient as a criterion for stopping of training. All three algorithms provided good results. The best value of Spearman's correlation coefficient between the value of the concreteness rating and its estimate was 0.906, which exceeds the values achieved in previous works.

Keywords: concreteness rating; abstractness; neural networks; fastText; word co–occurrence; English

1 Introduction

The issue of word concreteness has been the focus of attention of many academic disciplines for several decades. It is extensively studied in linguistics, psychology, psycholinguistics, medicine, neurophysiology, philosophy, and pedagogy [1]. The way abstract and concrete concepts are represented is a fundamental problem that has been debated in psychology, psycholinguistics, and neuro–physiology. A comprehensive review article [2] notes that the problem of representing abstract concepts is a crucial challenge for any theory of cognition.

Dictionaries with concreteness ratings of words are used to investigate this problem. For example, there are two large dictionaries of the English and Dutch languages created on the basis of native speakers‘ responses. Each of them includes approximately 40 thousand words [3, 4]. The dictionaries for other languages are tens of times smaller, specifically the Russian dictionary contains only 1000 words [5].

There are different approaches to the definition of abstractness and concreteness. It can be defined as (1) general, generic, not specific, and (2) lacking sense experience [6]. According to [6], nouns are considered concrete if they denote people, places and things and refer to a perceptible entity. If they cannot be experienced by our senses, they refer to more abstract concepts. Similar view on abstractness/concreteness is presented in [7], that is, abstract nouns are those that do not have denotata in real physical world and cannot be percepted. Criteria and norms that allow one to refer a word to an abstract or concrete concept are of great value for cognitive science.

However, instructing and asking participants of the experiments to validate the developed norms is a time– and labour–consuming process. Though there were some advances in this field such as Mechanical Turk, a service that allows collecting and processing data, as well as predicting values of concreteness. Using experts’ responses is still not an easy task. It is proved by the fact that the largest concreteness dictionaries provide ratings for a relatively small number of words. Creation of large text corpora and development of machine learning methods can contribute much to the solution of this problem.

One of the possible options is extrapolation of ratings obtained by expert assessment to a wider range of words. The degree of usefulness of such extrapolated ratings depends on how effectively the extrapolation procedure is realised. Extrapolated ratings are most useful when only small datasets of human judgments are available. Developing methods that allow for high–quality extrapolation from actual human judgments is a sort of breakthrough [8].

As stated above, it is too expensive and time–consuming to conduct experiments when experts determine concreteness ratings of large number of words (tens of thousands and more). Any corpus created using human ratings would be relatively small. However, large corpora are needed to solve practical tasks as the larger the corpus is, the more reliable are its data. The study objective is to develop a computational model for prediction of concreteness ratings. Using the model will be more efficient than conducting the experiments as it will allow one to obtain more trustworthy ratings in far lesser amount of time.

Three algorithms for estimating concreteness ratings of English words are compared in our work. These algorithms differ by the type of the used vector representation of words. Two of these algorithms have already been used to estimate concreteness ratings. The third algorithm that uses data on co–occurrence of the target words with the context ones is proposed for the first time in this

article. Therefore, we briefly discuss the reasons that make the introduced algorithm effective.

2 Related Works

Most studies on extrapolation of expert estimates and automatic expansion of the dictionary have been performed for the English language. The main idea and the study stages are as follows:

(1) A set of words with expert ratings of the degree of concreteness/abstractness is selected; some of the words are used to train the extrapolation method and the rest ones are used for testing.

(2) Words are represented by vectors in some semantic space.

(3) Some extrapolation method is applied to the data.

(4) The estimates obtained on the test set of words are compared with the expert ones.

Dictionaries presented in [3, 9] are selected as a set of words with expert ratings for the English language. In early studies, LSA was chosen as the semantic space; in recent works, a skip–gram model has been used for such purposes. Different types of Regression Models (SVM, neural networks, etc.) are often used as methods of approximation. Spearman‘s or Pearson’s correlation coefficient between ratings of concreteness and their estimates is utilized as accuracy measure of the model in most works. Table 1 summarizes the research results for the English language. The table includes papers in which correlation coefficients of 0.7 and higher were obtained.

Comments to the table:

(1) Some papers presented in Table 1 provide comparison of various methods. In this case, the best method is put in bold.

(2) Some papers use a very small set as a teaching one (the core). In this case, the number of concrete and abstract words are shown in parentheses in the “volume” column.

(3) The employed type of correlation coefficient is shown in the last column. Some papers use a binary classification (concrete/abstract words) instead of ranking. In this case, the value of the accuracy parameter is calculated instead of Spearman’s or Pearson’s correlation coefficient.

(4) The neural network was trained on sentences, not on separate words in [10] (the 7th row in the table). They used 800,000 sentences that contain 2580 words rated as abstract or concrete.

Table 1

Related works and obtained results for English

Paper, year	Corpus	Semantic space	method	volume train/test	correlation
[11], 2011	[9], English	LSA	A step-wise regression analysis	3,521 67%/33%	0.802 (Pearson)
[12], 2011	[9], English	LSA	Cosine similarity	4,295 0.9% (20-20) / 50%	0.822 (Spearman)
[13], 2013	[9], English	vector space representations from [14]	logistic regression classifiers	2,450 98%/2%	0.76 (accuracy)
[15], 2015	[3], English	LSA, topic model, a hyperspace analogue to language (HAL)-like model, a skip-gram model.	**k-nearest neighbours,** random forest	37,058 25%/75%	0.796 (Pearson)
[8], 2017	[3], English	a skip-gram model	step-wise regression model	37,058 50%/50%	0.829 (Pearson)
[16], 2018	[3], English	a skip-gram model	algorithm SentProp [17]	14,329 0.2%(15-15)/99.8%	0.70 (Spearman)
[10], 2018	English Wiki-pedia[1]	GloVe	Naive Bayes, Nearest neighbor, **RNN**	2580 81%/19%	0.740 (Pearson)
[18], 2018	[3,9], English	fasttext	**SVM**, feedforward networks	22,797 67%/33%	0.887 (Spearman)
[19], 2019	[3,20,21], English	fasttext	SVM	32,783 [3] / 2,005 [20,21]	0.902 (Pearson)

Now we note some results obtained in the above–mentioned works that were not included in the table. One of the first papers where high results were obtained is [22] (the Spearman's correlation coefficient is 0.64). The study [23] stands apart from the rest ones since it carries out extrapolation not within one language but between languages using a multilingual skip–gram model. In this case, the extrapolation method is trained on the full set of available data from one language. It is stated in [23] that the data were extrapolated on 77 languages; however, the

[1] English Wikipedia, May 2017 dump.

data on all languages are not presented. When extrapolating estimates from English to Dutch, Pearson's correlation coefficient with the expert estimates in Dutch from [4] equaled 0.76. One of the observations described in [23] is that more frequent nouns and verbs are less concrete in both English and Dutch.

Besides English and Dutch, both experts ratings and automatically generated ones were used to create dictionaries for other languages. For example, automatically obtained ratings for Chinese, Persian and Russian are presented in [24, 25] and [5, 26], respectively. If the algorithm starts with a small core (see papers [16, 12]), the question arises about selecting words for the core. The core of a fixed size in [16] included most frequent and most concrete and abstract (according to the expert ratings) words. The core of a fixed size in [12] contains 40 words. It is formed iteratively starting from an empty set and sequentially adding words that are in best correlation with abstract and concrete words from the training set. It is shown that if the core is expanded to 100 words, the Pearson's correlation coefficient on the test set will decrease.

3 Data and Method

The BWK base [3] was used as a source of ratings. This base provides concreteness ratings for about 40,000 words and word combinations. To test the trained models, we also utilized concreteness ratings from the MRC database [9], the Toronto Word Pool datasets [20] and the base created by Paivio, Yuille and Madigan [21] that provide concreteness ratings for 4239, 1093 and 925 words, respectively (the bases will be further abbreviated as MRC, TWP and PYM).

We use vector representations of words that have been developed within the framework of distributive semantics. The distributive semantics approach assumes that there is a correlation between distributional similarity and meaning similarity [27, 28, 29]. Currently, the most widely used methods are based on vector models of neural networks [30]. However, simpler representations based on explicit word vectors are also applied (see overviews [31]).

The first of the compared methods of word representations uses word embedding algorithms. Good reviews of word embedding methods can be found, for example, in [31, 32]. One of the main results in this area was described in the article [30] that introduced the word2vec model. Employing stochastic algorithms for learning artificial neural networks, the authors managed to obtain low–dimensional (with a dimension of 250-300) vector representations of words, which also implemented various semantic relations between them. One more significant achievement in this area was the fastText algorithm proposed in [33]. Combined usage of the word2vec model and subword information significantly reduces the time of model training and provides better result. The authors of [34] have granted free access to

four types of sets of pre–trained vectors (the dimension of word representation is 300). The sets differ by the source on which they were trained (Wikipedia 2017, UMBC webbase corpus or Common Crawl) and by whether subword information was used or not. Following the recommendation given in [19], we use vectors that do not include subword information.

Pre–trained fastText vectors have already been used in many studies on estimation of concreteness ratings. For example, two recent studies [18, 19], in which the highest results in the accuracy of concreteness rating estimation were obtained (see Table 1), use the fastText vectors as input data.

The second of the compared algorithms employs explicit word vectors. We use vector representations based on co–occurrence with the most frequent words (CFW). The CFW method is described, for example, in [35, 36]. The CFW method was applied in [37] to estimate the concreteness ratings of Russian words. In accordance with this approach, the target word is represented by a frequency vector of bigrams that include the target word and one of the context words. The CFW method uses a given number of the most frequent words as context words.

In our work, we use unigram and bigram frequency data extracted from the Google Books Ngram corpus [38]. The English (Common) subcorpus of Google Books Ngram includes texts of 16.6 million books published between 1470-2019 that contain approximately 2 trillion words. Currently, it is the largest corpus of the English language. Since we use the GBN corpus data, to make the list of context words, we selected 20,000 words that were most frequently used in GBN between 1900-2019.

Frequencies of combinations of each target word with each context word were extracted from the corpus (if some word combination was absent from the corpus, the corresponding frequency was considered equal to 0) As two types of bigrams are possible (with the target word in the first (Wx) and second places (xW)), we obtain two vectors with a dimension of 20,000. The last step is concatenation of these two vectors. Thus, a vector with a dimension of 40,000 is obtained for each target word. Besides ordinary bigrams, which are pairs of consecutive words, GBN contains information on syntactic bigrams [39]. We compared types of vector representation obtained using data for both ordinary and syntactic bigrams.

As a rule, the resulting vectors are very sparse (contain a large number of zeros); however, they carry all information about the co–occurrence of the target word with the most frequent ones. The drawback of this method is high dimension of the resulting vector representation, which can cause significant problems in the process of training neural network models (especially for fully connected networks) and lead to overfitting. Therefore, if this type of word representation is used, it is important to ensure good regularization of the model during the training process.

The third of the compared algorithms is proposed in this article for the first time. We also use explicit word vectors. The scheme of the proposed algorithm is analogous to the CFW method with the only difference that we use functional words (not always the most frequent ones) as context words. The proposed algorithm will further be called the CSW algorithm (co–occurrence with stop–words). It was abbreviated as CSW for the following reason. If we abbreviated it as CFW (co–occurrence with functional words), this could lead to misunderstanding as the abbreviation CFW already exists and is mentioned in this paper. Therefore, we replaced the letter F by S, where S refers to stop–words. By stop–words we understand functional words presented in [40]. We borrowed the list of 307 functional words from [40]. It includes articles, conjunctions, particles, prepositions, as well as numerals, auxiliary verbs, some adjectives, pronouns, etc.

We used the GBN corpus to extract frequencies of bigrams that include the target words and one of the functional words. Thus, we obtained a vector representation of dimension 614 for each target word (taking into account bigrams of the type Wx and xW).

The pre–trained fastText vectors can be directly fed into the neural network input; however, when using explicit word vectors, appropriate preprocessing is required. The first problem is a large range of change in bigram frequencies. For example, a set of vectors that we used contains frequency values from 40 to $1.8 \cdot 10^{10}$. The second problem results from the fact that values of absolute frequencies depend on a corpus size; and if it is required to use the obtained model on other data, the vectors need to be normalized. Based on the experience of previous works (see, for example, [41]), two preprocessing methods were chosen.

The first one proposed in [42] assumes that frequency values are used to calculate the corresponding Pointwise Mutual Information values.

$$PMI_{i,j} = \log_2 \frac{f_{i,j}}{f_i f_j} \tag{1}$$

Here f_i is the relative frequency of the i-th target word, f_j is the relative frequency of the j-th context word, f_{ij} is the relative frequency of the bigram in-cluding the i-th target word and the j-th context word. On the one hand, PMI is composed of relative values and does not depend on the size of the employed corpus; on the other hand, it provides compactification of the dynamic range due to the presence of a logarithm.

The second considered preprocessing method is taking a simple logarithm of frequency vectors. This technique also allows one to reduce the dynamic range of the vector input values; however, it does not eliminate the dependence on the size of the em-ployed corpus. Nevertheless, this preprocessing method has shown good results in several tasks [41]. To perform preprocessing correctly when frequency value equals zero, 1 is added to the frequencies before taking the logarithm:

$$\log_2\left(F_i+1\right) \tag{2}$$

where F_i is the frequency of the bigram at the i-th position of the input vector.

The traditional fully connected feedforward network [43] was chosen as a model that solves the problem of estimating concreteness ratings of words. It consisted of 4 hidden layers; each of them contained 128 neurons. Each neuron in the hidden layer used ELU [44] as activation function. The output layer contained 1 neuron with an identically linear activation function.

Despite the fact that we used the same neural network architecture in all three cases, the number of weights in the network is significantly different in each case. The number of variable network parameters was about 5.1 million for the CFW method. Due to the large dimension of the input vector (D = 40,000), almost 99.5% of all free parameters of the model were concentrated in the input layer. Therefore, much attention was paid to regularization when training this model. Regularization was also used for the models based on the rest two methods (the dimension of the input vectors equaled 300 (D = 300) and 614 (D = 614), respectively); however, its impact on the training process was significantly lower.

The main regularizer was the dropout layer placed between the input and the first hidden network layer. Stochastic disabling of connections between neurons provides regularization and prevents overfitting of the neural network [45]. The regularization parameter of 0.3 was chosen for the model based on the CFW method. Thus, only random 70% of all connections of the layer were used and corrected at each training iteration. The dropout parameter was significantly lower (0.1) for the rest two methods. Beyond that, L1–regularization of all hidden layers was additionally employed when the CFW method was used. This allowed us to obtain a sparser representation as well as to reduce the tendency of this model to overfit.

The mean square error (MSE) between the target value of the concreteness rating and the resulting network estimate was chosen as a loss function for all three types of models. The model was trained based on stochastic gradient descent by the Adam [46] method. At each training iteration, random 128 examples from the training sample formed a training batch, the root mean square error of which was minimized by the network. Simultaneously, a similar batch of the same size was generated from the test sample for network validation. When the target loss did not decrease by more than 10% during 1,000 iterations of updating the weights, we artificially reduced the learning rate parameter [47]. Each time this condition was met, the learning rate was reduced by half. This allowed improving the network results obtained at the last stages of its training when the values of the target loss function have practically not changed.

In addition to the loss function, Spearman's correlation coefficient was chosen as an additional metric and calculated on the test sample during the training process.

Spearman's correlation coefficient on the entire test sample is calculated significantly longer in comparison with the time spent on the forward and backward signal propagation through the network. Since this metric had to be read after each iteration of the network weight adjustment, its stochastic version was implemented.

Spearman's correlation coefficient was calculated using random 2048 samples from the test sample. At that, such a truncated metric turned out to be a representative estimate of the Spearman's correlation coefficient calculated for the entire test sample. This metric was used to control the overfitting of the network, as well as a criterion for stopping the training process. After the values of stochastic Spearman's correlation coefficient reached a plateau, the training stopped after 1000 updates of the weights. The network weights corresponding to the highest value of this metric were further used to test the training results.

The PyTorch [48] automatic differentiation library was used as a framework for training the neural network model.

Thus, we trained and tested 10 models in total. They were represented by two models for the case of using fastText vectors trained on Wikipedia and CommonCrawl, four models for each case of employing the CFW and CSW methods (for ordinary and syntactic bigrams, and two types of input data preprocessing).

As it was stated above, in many papers, the obtained accuracy is estimated by calculating Spearman's or Pearson's correlation coefficients between a concreteness rating and its estimate. Some papers (see, for example, [19]) use Kendall's correlation coefficient. To simplify comparison with prior works, in most cases, we provide values of the three coefficients. In our opinion, the use of Spearman's correlation coefficient is the most justified in this case since its employment is not associated with certain assumptions about distribution of the analyzed data [49].

4 Results

As mentioned in the previous section, 20% of words presented in the [3] database were selected for testing. These words were not used for training. The selection of words for the test sample was carried out randomly; and the words included in it have the same frequency distribution and part–of–speech distribution as the words in the training sample. The test sample included 7406 words for the models that use bigram frequencies extracted from GBN as input data and 6815 words for the models that employ pre–trained fastText vectors. Here, 7406 words are 20% of 37,030 words that are found both in the BWK base and GBN; and 6815 words are 20% of 34,076 words that are found in the BWK base and have fastText vectors.

As an example, Figure 1 shows bidimensional distribution of concreteness rating values and their CFW (ordinary bigrams, PMI) estimates. The figure illustrates that the quality of rating estimation is quite high.

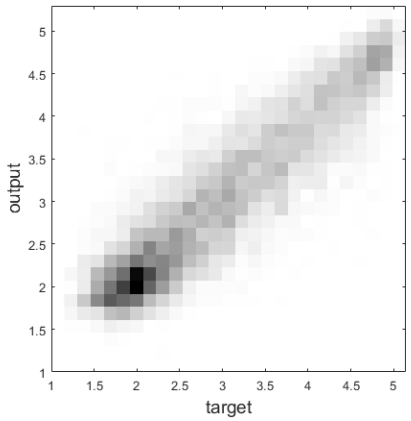

Figure 1

Bidimensional distribution of concreteness rating values and their estimates obtained using CFW (ordinary bigrams, PMI)

The values of the correlation coefficients between the values of the concreteness rating and its estimation were calculated on the test set for each of the 10 models. The results obtained using the CFW and CSW methods are shown in Table 2.

Table 2

Pearson's (r), Spearman's (ρ) and Kendall's (τ) correlation coefficients between the concreteness rating values and their estimates obtained using the CFW and CSW methods (for the BWK dataset)

Set of Vectors		PMI	$Log_2(1+x)$
CFW, ordinary bigrams	r	0.899	0.901
	ρ	0.884	0.884
	τ	0.702	0.701
CFW, syntactic bigrams	r	0.902	0.902
	ρ	0.888	0.887
	τ	0.706	0.704
CSW, ordinary bigrams	r	0.890	0.883
	ρ	0.875	0.868
	τ	0.688	0.680
CSW, syntactic bigrams	r	0.890	0.878
	ρ	0.873	0.861
	τ	0.686	0.671

It is obvious that the CFW method has a slight advantage over CSW. It should be noted that the results obtained using ordinary and syntactic bigrams almost do not differ. The two data preprocessing methods also provided approximately the same results when we used the CFW method. When the CSW method was employed, the use of PMI vectors significantly improved the accuracy. These results differ form ones obtained in [41] and shows that solving different tasks may require different preprocessing methods.

Table 3 shows the results of testing the models based on fastText vectors. Better results are obtained when using vectors trained on the CommonCrawl corpus.

Table 3

Pearson's (r), Spearman's (ρ) and Kendall's (τ) correlation coefficients between the concreteness rating values and their estimates for the [3] dataset using fastText vectors

Set of Vectors	r	ρ	τ
CommonCrawl	0.916	0.906	0.729
Wikipedia	0.901	0.893	0.710

For each of the three methods, we selected the variant that provided best results. For the CFW method, it we used syntactic bigrams and PMI (=0.888); for the CSW method, we employed ordinary bigrams and PMI (=0.875); and for the method employing fastText vectors, we utilized vectors pre–trained on the CommonCrawl corpus (=0.906). For each of the described cases, the neural model was trained 10 times and tested. Standard deviation of Spearman's correlation coefficient between the concreteness rating and its estimate was $4 \cdot 10^{-3}$ for the CFW method, $3.5 \cdot 10^{-3}$ for the CSW method, and $1.7 \cdot 10^{-3}$ for the fastText method. Comparing the obtained values with those shown in Tables 2,3, one can see that the differences in accuracy between the three methods are not large, however, they are statistically significant.

Ideally, comparative testing of different methods should be carried out using corpora of the same size. Unfortunately, in practice, one has to use available tools and datasets. In our case, the size of the CommonCrawl corpus that was used to obtain fastText vector representation is about 3 times smaller than that of the English (common) subcorpus of GBN employed to obtain vectror representations by the CFW and CSW methods. However, this does not cause difficulties in determining which of the compared methods showed the highest accuracy. The highest result was obtained using the pre–trained fastText vectors. If we used a corpus which 3 times exceeds the size of CommonCrawl, we would probably expect even more increase in accuracy.

Now we consider how estimation accuracy of concreteness rating depends on word frequency. To perform a comparative analysis of the three algorithms, we select the variant that provides the highest values. These variants are the use of syntactic bigrams and PMI (for the CFW method) and the use of fastText vectors obtained on the CommonCrawl corpus (for the CSW method).

The following approach was used to analyze the dependence of accuracy on frequency. We sort the words in the test sample in descending order of frequency. After that, we calculate Spearman's correlation coefficient between the rating values and its estimates in a sliding window with a length of 1000. Each position of the window defines a certain frequency range. We take the geometric mean of frequency of words that fell into the sliding window for visualization. The obtained results are shown in Figure 2.

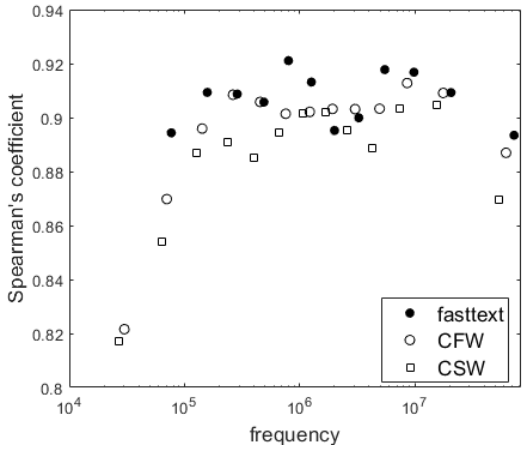

Figure 2

Dependence of Spearman's correlation coefficient between the values of concreteness rating and their estimates on word frequency

The figure shows that the frequency dependence is weak in a wide frequency range (from 105 and higher). At that, some advantage of estimates obtained using fastText vectors is observed. The accuracy of the estimation obtained using the CFW and CSW methods starts decreasing with frequency decrease. It is a complicated task to analyze the accuracy of the fastText method for this frequency range since there are few rare words in the corresponding test sample.

Table 4

Spearman's correlation coefficients (ρ) between the values of the concreteness rating and their estimates for different parts of speech

Method	NOUN	ADJ	VERB
CFW	0.899	0.774	0.804
CSW	0.886	0.733	0.770
fastText	0.912	0.785	0.826
percentage of case, %	55	21	15

Table 4 presents data on the accuracy of concreteness rating estimation for each of the main parts of speech separately. For each of the three methods, we choose the variant mentioned above when we considered the impact of frequency on the estimation accuracy. The last line in the table shows the percentage of words related to one or another part of speech in the test sample. The highest accuracy of concreteness rating estimation is achieved for nouns. The Accuracy for verbs and even more so for adjectives is lower. One of the reasons is that there are significantly more nouns in the training set than verbs and adjectives. Therefore, the model was better trained for nouns.

Let us compare the level of accuracy achieved by us with the results obtained in previous works. As far as we know, currently, the highest accuracy in estimating the specificity rating has been achieved by the authors of [18, 19]. Testing is carried out on the BWK data in [18]. This work employs fastText vectors as input data; and comparative testing of two regression algorithms is carried out using the SVM method and the feed–forward neural network. The obtained values of the Spearman's correlation coefficient between the values of the concreteness rating and its estimate for these algorithms are 0.887 and 0.879, respectively. It should be mentioned that the second of the algorithms described in [18] is similar to one of the methods that we compare in our work (it uses the fastText vectors); however, the level of accuracy we obtained is noticeably higher (0.906 versus 0.879). It can be assumed that this is primarily due to the use of a different criterion for stopping of training, as well as the difference in the employed regularization methods.

Table 5
Pearson's (r), Spearman's (ρ) and Kendall's (τ) correlation coefficients between the concreteness rating values and their estimates for different datasets

Method		TWP	PYM	MRC
CFW	r	0.910	0.915	0.900
	ρ	0.875	0.926	0.901
	τ	0.698	0.764	0.722
CSW	r	0.909	0.916	0.891
	ρ	0.886	0.913	0.895
	τ	0.710	0.747	0.711
fastText	r	0.890	0.916	0.892
	ρ	0.878	0.920	0.896
	τ	0.696	0.745	0.712
fastText+SVM [19]	r	0.881	0.902	-
	ρ	-	-	-
	τ	0.698	0.741	-

The test results are given for the MRC database, as well as for TWP and PYM datasets in [19]. We select words from these three datasets, which are also present in the test sample. Table 5 shows the values of the correlation coefficients

between the estimates obtained by the neural network and the concreteness ratings extracted from the TWP, PYM, and MRC. Similar to [19], we compare the rating values not with the target values extracted from BWK but with the values of the ratings in these three datasets.

The table shows that all three methods under consideration provide better results than those described in [19]. Charbonnier and Wartena [19] also raised an important question about the limit of the achievable accuracy in estimating the concreteness ratings. Therefore, the authors [19] analyzed the level of correlation between the rating values given in different datasets. Table 6 shows the values of the correlation coefficients between the concreteness rating values given in the BWK and the ratings of the same words presented in the MRC, TWP and PYM databases. One can see that we obtained the values of the correlation coefficients between the target value of the rating and its estimation (see Table 5) that almost reach the values shown in Table 6. In many cases, the difference is only a few thousandths.

Table 6

Pearson's (r), Spearman's (ρ) and Kendall's (τ) correlation coefficients between the concreteness rating values given in BWK and other datasets

Dataset	r	ρ	τ
TWP	0.913	0.899	0.736
PYM	0.936	0.932	0.770
MRC	0.919	0.921	0.748

This seems surprising since the models were trained only on BWK ratings. To understand the reason, we select those words from the test sample that are also found in one of the other datasets (TWP, PYM or MRC). For example, the test sample contains 199 words that are also present the TWP dataset. In 114 cases of 199 (or in 57.3% of all cases), the estimate obtained by the neural network deviates from the target value in the same direction as the value of the concreteness rating in TWP (here we use estimates obtained using the CSW, PMI methods and ordinary bigrams). A priori, it is natural to assume that deviations of the estimation of the concreteness rating from the target value upward and downward are equally probable. If this hypothesis is correct, then the number of cases where the estimates obtained by the neural network lie closer to the values from the TWP than the target values (extracted from the [3] base) should obey the binomial distribution with the parameter 0.5. It is easy to calculate that the p–value for this case is 0.0234. The test sample contains 166 words that are also included in the PYM dataset. The estimates of concreteness rating of 93 words from 166 (56% of all cases) are closer to the PYM ratings than the target values. In this case, the p–value is 0.070. Finally, the MRC base contains 773 words that are also present in the test sample. In 463 cases (56.4% of all cases) the estimates deviate from the target values so that their difference from the target values given in MRC decreased. The p–value for this case is $2.08 \cdot 10^{-4}$. Thus, at any reasonable

level of significance, the null hypothesis that the estimate is equally likely to deviate from the target value upward and downward should be rejected.

Thus, although the neural network was trained only on BWK data, due to the ability of the neural network to generalize, the rating estimates often deviate from the target ones in such a direction as to approach the rating values from other datasets. That is, the neural network seeks to "correct" errors occurred during rating estimation performed by individual research groups.

6 Interpretation of Results

It is a surprising fact that using CSW provides accuracy that is slightly lower than that obtained by the other two methods. Indeed, less information is fed to the input of the neural network in this case than employing the other two considered algorithms. From the utilized list of functional words, 299 (or 97.4%) are also included in the list of 20,000 most frequent words applied in the CFW method. Adding 19,700 more context words to the list of context words allows us to raise Spearman's correlation coefficient of the concreteness rating and its estimate from 0.875 to only 0.888. In this section, we will try to explain why it becomes possible to achieve high accuracy in estimating the concreteness rating using the CSW method.

We repeated the calculation of the ratings, disconnecting one of the inputs of the neural network in turn. This was done by feeding the corresponding input zero values for each target word. Then, we calculated the increments of Spearman's correlation coefficient by formula 3:

$$\Delta\rho^{(i)} = \rho^{(i)} - \rho \tag{3}$$

Here ρ is Spearman's correlation coefficient for a network using all inputs, and $\rho^{(i)}$ is Spearman's correlation coefficient for a network with the disabled i–th input. Notice that the more useful a type of bigram is for determining the concreteness rating, the more significant the drop in Spearman's correlation coefficient will be when the corresponding input is turned off.

At the next stage, we sorted all bigrams in the descending order of the $\Delta\rho^{(i)}$ increments. The words (more precisely, the construction with the words) that have the greatest influence on the concreteness ratings of the target words were at the top of the list. As the words referring to different parts of speech may have different "influential" context words, we performed the described calculations for each of the studied part of speech.

We analysed 614 contexts the studied words appear in and ranged the context words (Wx, xW types) according to their contribution to the concreteness ratings.

We formed 3 lists of words. The first list included ranged context words that influence the concreteness ratings of nouns, the rest ones consisted of words that influence the ratings of verbs, adverbs, and adjectives, respectively. The task is to describe the most typical group of words form the list without detailed semantic analysis, though we give some clues why the studied words are in the list taking certain place.

The first group we analysed was the list of combinations of functional words with nouns. The most "influential" word in this group is the indefinite article *a* ("a+X") that is usually used with concrete countable nouns in the considered construction. The definite article *the* takes the third place in the rating that can be used both with abstract and concrete nouns, however, we can say that nous preceded by *the* are often more concrete than those with the zero and indefinite articles.

The third and thirteenth top constructions are "X+of" and "of+X", respectively. They form the genitive construction that usually shows relations between two nouns, such as mereology, taxonomy, valency, etc. Used both with concrete and abstract nouns (*out of curiosity*), we may hypothesize that this construction is more typical of concrete nouns. The top construction "X+from" can be compared to the genitive construction considering mereology, it describes the part divided from the whole. It resembles extraction of something or somebody from something. We suppose that it is more often used with concrete nouns in the studied construction. The construction "X+with" often describes the whole with the added part. It may be used with concrete/abstract nouns and in set expressions (*in love with somebody),* however, concrete nouns are more expected to be used in this structure. The preposition on ("X+on") assumes something/somebody locating on something/somebody, i.e contact between the figure and the ground [50]. It is used both with concrete and abstract nouns (*shame on yo*u), however, concrete sense occurs more often [51]. Primarily function of all prepositions is to describe spatial relationships between concrete nouns though abstract uses are also common. The considered top list prepositions are *beside* ("X+beside"), *within* ("X+within"), *among* ("X+among"), *onto* ("X+onto"), *into* ("into+X"), *towards* ("towards+X") etc. Their contribution to abstract/concrete correlation is valuable.

Besides prepositions, the list of context words included adjectives like *each* ("each+X") and *every* ("every+X") denoting "every one of two or more considered individually or one by one", "being one of a group or series taken collectively" (https://www.dictionary.com). Such words are usually used with countable nouns that denote concrete nouns. Thus, they provide higher correlations in abstractness estimation.

Numerals also have contribution to the ratings. The structures like "two/three/four five+X" and "X+two/three/four/five" with the latter ranked higher in the list. The word first is at the top of the list.

Possessive pronouns and demonstrative adjectives ("my/their/our/your+X" and "this/these+X"), are also among fifty most "influential words". Possessive

pronouns refer to something that we have or that relate to us (in a wide sense). It seems that we possess something visible and concrete though we can, for example, feel something and describe as "my feeling". If considering theme/rheme relations, nouns determined by possessive pronouns are more concrete in the contexts than undetermined ones. Demonstrative adjectives refer to different type of objects, but the grammatical structure implies that *this* refers to one object and *these* – to several objects. *This* can combine with both concrete and abstract nouns depending on the context. However, these usually refer to several concrete objects.

The quantifiers *much* ("much+X") and *many* ("many+X") are in the middle of the list. *Much* is used with singular uncountable nouns that are often abstract nouns; *many* is used with plural nouns that are usually concrete. Therefore, they contribute much to concreteness ratings estimation.

There are conjunctions ("and+X", "or+X"), reflexive pronouns ("ourselves+X"), auxiliaries ("X+will"). They are less "influential" considering concreteness ratings.

When analysing concreteness ratings of adjectives, we should bear in mind nouns because adjectives modify nouns. If the adjective occurs with more abstract noun than usual, it is an indicator of metaphoricity [52], therefore, its sense becomes more abstract. Thus, the construction "a+adjective" implies that there is some noun behind. If we consider nominal predicates, the modified noun is before the adjective. The ranged context words for adjectives showed some correlation with the ranged list created for nouns. For example, the articles ("a/the+X"), demonstrative adjectives ("this/these+X"), quantifiers ("X+many") are also at the top of the list. However, detailed analysis lies in the linguistic domain.

We also ranked the context words that influence the correlation of verbs concreteness. Among the most "influential" ones are reflexive pronouns ("X+themselves, ourselves, herself", "yourself+ X"), and adverbial modifiers ("X+again", "already+X").

Conclusions

We compared three algorithms for estimating concreteness ratings of English words. To train and test the models, we used a number of freely available databases that contain concreteness ratings [3, 9, 20, 21].

Spearman's correlation coefficient between the concreteness rating and its estimate of 0.906 was obtained on the test sample of words included in the BWK database [3]. Even higher correlation values were obtained for high–frequency words included in the [21] dataset and the MRC Psycholinguistic Database. The achieved level of accuracy exceeds the values obtained in previous works.

Increase in accuracy became possible due to some improvements of the training process of the models. The most significant improvement was the use of stochastic

Spearman's correlation coefficient that was employed as the second metric in the training process. The value of this metric was used as a criterion for stopping the training process, as well as for choosing the best iteration.

The comparison of the three tested algorithms shows that each of them has its own advantages. The algorithm that uses fastText as input data showed the highest accuracy and can be used to extrapolate concreteness ratings to a wide range of words based on synchronous data. The other two algorithms showed slightly lower accuracy. However, their advantage is easy adaptation to diachronic data, for example, when using large amount of data from the Google Books Ngram corpus. A recent work [16] showed that the values of the concreteness rating of some words change significantly over time. This phenomenon is of great interest and needs further study. The CSW method seems especially promising for diachronic studies since combinations with functional words are usually quite frequent. Besides possible practical applications of the CFW method, the fact that its accuracy is practically not inferior to the accuracy of the other two considered methods is of interest from the point of view of theory.

Another advantage of the algorithms using explicit word vectors is the ease of interpretation of the obtained results. If a model employing word embeddings is a black box for models based on explicit word vectors, we can determine which combinations occurring in the corpus increase or decrease the concreteness estimates.

The results obtained in this work can be used to create large dictionaries with concreteness ratings of words and other semantic and psychological variables, which is important for many practical applications.

Acknowledgement

This research was financially supported by Russian Foundation for Basic Research (grant 19–0700807) and by the Kazan Federal University Strategic Academic Leadership Program.

References

[1] Solovyev, V. Concreteness/Abstractness Concept: State of the Art. In: Advances in Cognitive Research, Artificial Intelligence and Neuroinformatics, pp. 275-283, Springer International Publishing, Cham, 2021

[2] Borghi, A., Binkofski, F., Castelfranchi, C., Cimatti, F., Scorolli, C., Tummolini, L. The challenge of abstract concepts, Psychological bulletin, V. 143, N. 3, 2017, pp. 263-292

[3] Brysbaert, M., Warriner, A. B., Kuperman, V. Concreteness ratings for 40 thousand generally known English word lemmas, Behavior research methods, V. 46, N. 3, 2014, pp. 904-911, doi: 10.3758/s13428-013-0403-5

[4] Brysbaert, M., Stevens, M., De Deyne, S., Voorspoels, W., Storms, G. Norms of age of acquisition and concreteness for 30,000 Dutch words, Acta Psychologica, V. 150, 2014, pp. 80-84, doi: 10.1016/j.actpsy.2014.04.010

[5] Solovyev, V. D., Ivanov, V. V., Akhtiamov, R. B. Dictionary of Abstract and Concrete Words of the Russian Language: A Methodology for Creation and Application, Journal of Research in Applied Linguistics, V. 10, 2019, pp. 218-230, doi: 10.22055/rals.2019.14684

[6] Spreen, O., Schulz, R. W. Parameters of abstraction, meaningfulness, and pronunciability for 329 nouns, Journal of Verbal Learning and Verbal Behavior, V. 5, N. 5, 1966, pp. 459-468, doi: 10.1016/S0022-5371(66)80061-0

[7] Schmid, H.-J. English Abstract Nouns as Conceptual Shells, De Gruyter Mouton, Berlin, Boston, 2012, doi: 10.1515/9783110808704

[8] Hollis, G., Westbury, C., Lefsrud, L. Extrapolating human judgments from skip-gram vector representations of word meaning, Quarterly Journal of Experimental Psychology, V. 70, N. 8, 2017, pp. 1603-1619, doi: 10.1080/17470218.2016.1195417

[9] Coltheart, M. The MRC psycholinguistic database, The Quarterly Journal of Experimental Psychology Section A, V. 33, N. 4, 1981, pp. 497-505, doi: 10.1080/14640748108400805

[10] Rabinovich, E., Sznajder, B., Spector, A., Shnayderman, I., Aharonov, R., Konopnicki, D. and Slonim, N. Learning Concept Abstractness UsingWeak Supervision. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4854-4859, Association for Computational Linguistics, Brussels, Belgium, 2018, doi:10.18653/v1/D18-1522

[11] Feng, S., Cai, Z., Crossley, S., McNamara, D. Simulating human ratings on word concreteness. In: Proceedings of the 24th International Florida Artificial Intelligence Research Society, FLAIRS – 24, pp. 245-250, AAAI, Florida, 2011

[12] Turney, P., Neuman, Y., Assaf, D., Cohen, Y. Literal and Metaphorical Sense Identification through Concrete and Abstract Context. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 680-690, Association for Computational Linguistics, Edinburgh, Scotland, UK, 2011, https://aclanthology.org/D11-1063

[13] Tsvetkov, Y., Mukomel, E., Gershman, A. Cross-Lingual Metaphor Detection Using Common Semantic Features. In: Proceedings of the First Workshop on Metaphor in NLP, pp. 45-51, Association for Computational Linguistics, Atlanta, Georgia, 2013, https://aclanthology.org/W13-0906

[14] Huang, E., Socher, R., Manning, C., Ng, A. Improving Word Representations via Global Context and Multiple Word Prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 873-882, Association for Computational Linguistics, Jeju Island, Korea, 2012, https://aclanthology.org/P12-1092

[15] Mandera, P., Keuleers, E., Brysbaert, M. How useful are corpus-based methods for extrapolating psycholinguistic variables? Quarterly Journal of Experimental Psychology, V. 68, N. 8, 2015, pp. 1623-1642, doi: 10.1080/17470218.2014.988735

[16] Snefjella, B., Gnreux, M., Kuperman, V. Historical evolution of concrete and abstract language revisited, Behavior research methods, V. 51, N. 4, 2019, pp. 1693-1705, doi:10.3758/s13428-018-1071-2

[17] Hamilton, W. L., Clark, K., Leskovec, J., Jurafsky, D. Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 595-605, Association for Computational Linguistics, Austin, Texas, 2016, doi: 10.18653/v1/D16-1057

[18] Ljubeic, N., Fier, D., Peti-Stantic, A. Predicting Concreteness and Imageability of Words Within and Across Languages via Word Embeddings. In: Proceedings of The Third Workshop on Representation Learning for NLP, pp. 217-222, Association for Computational Linguistics, Melbourne, Australia, 2018, doi:10.18653/v1/W18-3028

[19] Charbonnier, J., Wartena, C. Predicting Word Concreteness and Imagery. In: Proceedings of the 13th International Conference on Computational Semantics - Long Papers, pp. 176-187, Association for Computational Linguistics, Gothenburg, Sweden, 2019, doi:10.18653/v1/W19-0415

[20] Friendly, M., Franklin, P., Hoffman, D., Rubin, D. The Toronto Word Pool: Norms for imagery, concreteness, orthographic variables, and grammatical usage for 1,080 words, Behavior Research Methods & Instrumentation, V. 14, 1982, pp. 375-399

[21] Paivio, A., Yuille, J., Madigan, S. Concreteness, imagery, and meaningfulness values for 925 nouns, Journal of experimental psychology, V. 76, N. 1, 1968, Suppl:pp. 1-25, doi:10.1037/h0025327

[22] Theijssen, D., van Halteren, H., Boves, L., Oostdijk, N. On the difficulty of making concreteness concrete, Computational Linguistics in the Netherlands Journal, V. 1, 2011, pp. 61-77

[23] Thompson, B., Lupyan, G. Automatic estimation of lexical concreteness in 77 languages. In: Proceedings of the 40th Annual Conference of the Cognitive Science Society (CogSci 2018), pp. 1122-1127, Cognitive Science Society, Madison, WI, USA, 2018

[24] Wang, X., Su, C., Chen, Y. A Method of Abstractness Ratings for Chinese Concepts. In: UKCI: UKWorkshop on Computational Intelligence, pp. 217-226, Springer International Publishing, 2018, doi: 10.1007/978-3-319-97982-3

[25] Dadras, P., Ramezani, M. CODAC: Concreteness Degree Auto-Calculator of Persian Words, International Journal of Computer Science and Information Security (IJCSIS), V. 15, N. 5, 2017, pp. 64-72

[26] Solovyev, V., Ivanov, V. Automated Compilation of a Corpus-Based Dictionary and Computing Concreteness Ratings of Russian. In: Speech and Computer, pp. 554-561, Springer International Publishing, Cham, 2020

[27] Harris, Z. S. Papers in structural and transformational linguistics, Reidel, Dordrecht, 1970

[28] Weeds, J., Weir, D., McCarthy, D. Characterising Measures of Lexical Distributional Similarity. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pp. 1015-1021, COLING, Geneva, Switzerland, 2004, https://aclanthology.org/C04-1146

[29] Pantel, P. Inducing Ontological Co-Occurrence Vectors. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 05, pp. 125-132, Association for Computational Linguistics, USA, 2005, doi: 10.3115/1219840.1219856

[30] Mikolov, T., Chen, K., Corrado, G. and Dean, J. Efficient Estimation of Word Representations in Vector Space, arXiv preprint, arxiv:1301.3781, 2013, http://arxiv.org/abs/1301.3781

[31] Tang, X. A state-of-the-art of semantic change computation, Natural Language Engineering, V. 24, N. 5, 2018, pp. 649-676, doi:10.1017/S1351324918000220

[32] Tahmasebi, N., Borin, L., Jatowt, A. Survey of computational approaches to lexical semantic change detection. In: Computational approaches to semantic change, pp. 1-91, Language Science Press, Berlin, 2021

[33] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jgou, H. and Mikolov, T. FastText.zip: Compressing text classification models, arXiv preprint, arXiv:1612.03651, 2016, https://arxiv.org/abs/1612.03651

[34] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A. Advances in Pre-Training Distributed Word Representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018

[35] Xu, Y. and Kemp, C. A Computational Evaluation of Two Laws of Semantic Change, Cognitive Science, 2015

[36] Khristoforov, S., Bochkarev, V., Shevlyakova, A. Recognition of Parts of Speech Using the Vector of Bigram Frequencies. In: Analysis of Images,

Social Networks and Texts, pp. 132-142, Communications in Computer and Information Science, Springer International Publishing, Cham, V. 1086, 2020, doi: 10.1007/978-3-030-39575-9 13

[37] Solovyev, V. D., Bochkarev, V. V., Khristoforov, S. V. Generation of a dictionary of abstract/concrete words by a multilayer neural network, Journal of Physics: Conference Series, V. 1680, 2020, 012046, doi:10.1088/1742-6596/1680/1/012046

[38] Lin, Y., Michel, J.-B., Aiden Lieberman, E., Orwant, J., Brockman, W., Petrov, S. Syntactic Annotations for the Google Books NGram Corpus. In: Proceedings of the ACL 2012 System Demonstrations, pp. 169-174, Association for Computational Linguistics, Jeju Island, Korea, 2012, https://aclanthology.org/P12-3029

[39] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernndez, L. Syntactic Dependency-Based N-grams as Classification Features. In: Advances in Computational Intelligence, pp. 1-11, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013

[40] Hughes, J. M., Foti, N. J., Krakauer, D. C., Rockmore, D. N. Quantitative patterns of stylistic influence in the evolution of literature, Proceedings of the National Academy of Sciences, 2012, doi:10.1073/pnas.1115407109

[41] Savinkov, A., Bochkarev, V., Shevlyakova, A., Khristoforov, S. Neural Network Recognition of Russian Noun and Adjective Cases in the Google Books Ngram Corpus. In: Speech and Computer, SPECOM 2021, pp. 626-637, Lecture Notes in Computer Science, Springer International Publishing, Cham, V. 12997, 2021, doi: 10.1007/978-3-030-87802-3 56

[42] Bullinaria, J. A., Levy, J. P. Extracting semantic representations from word co-occurrence statistics: A computational study, Behavior Research Methods, V. 39, N. 3, 2007, pp. 510-526, doi:10.3758/BF03193020

[43] Haykin, S. Neural Networks: A Comprehensive Foundation, (2nd ed.), Prentice Hall PTR, USA, 1998

[44] Shah, A., Kadam, E., Shah, H., Shinde, S. Deep Residual Networks with Exponential Linear Unit, arXiv preprint, arXiv:1604.04112, 2016, http://arxiv.org/abs/1604.04112

[45] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, J. Mach. Learn. Res., V. 15, N. 1, 2014, pp. 1929-1958

[46] Kingma, D. P., Ba, J. Adam: A Method for Stochastic Optimization, arXiv preprint, arXiv:1412.6980, 2014, http://arxiv.org/abs/1412.6980

[47] You, K., Long, M., Jordan, M. I. How Does Learning Rate Decay Help Modern Neural Networks, arXiv preprint, arXiv:1908.01878, 2019, https://arxiv.org/abs/1908.01878

[48] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai J., Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems 32, pp. 8024-8035, Curran Associates, Inc., 2019

[49] Kendall, M. G., Stuart, A. The advanced theory of statistics. Inference and relationship. (3rd ed.), Griffin, London, 1961

[50] Jamrozik, A., Gentner, D. Making sense of the abstract uses of the prepositions in and on. In: Proceedings of the 36th Annual Conference of the Cognitive Science Society, pp. 2411-2416, Cognitive Science Society, 2014

[51] Steen, G. J., Dorst, A. G., Herrmann, J. B., Kaal, A. A., Krennmayr, T., Pasma, T. A method for linguistic metaphor identification. From MIP to MIPVU., Converging Evidence in Language and Communication Research, John Benjamins, V. 14, 2010

[52] Hill, F., Korhonen, A. Concreteness and Subjectivity as Dimensions of Lexical Meaning. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 725-731, Association for Computational Linguistics, Baltimore, Maryland, 2014, doi: 10.3115/v1/P14-2118

CoLI-Machine Learning Approaches for Code-mixed Language Identification at the Word Level in Kannada-English Texts

Shashirekha Hosahalli Lakshmaiah [1,a], Fazlourrahman Balouchzahi[2,b], Mudoor Devadas Anusha[1,c] and Grigori Sidorov[2,d]

[1] Department of Computer Science, Mangalore University Mangalore - 574199, India; [a] hlsrekha@mangaloreuniversity.ac.in, [c] anusha@mangaloreuniversity.ac.in

[2]Instituto Politécnico Nacional, Centro de Investigación en Computación, CDMX-07738, Mexico; [b] frs_b@nlp.cic.ipn.mx, [d] sidorov@cic.ipn.mx

Abstract: The task of automatically identifying a language used in a given text is called Language Identification (LI). India is a multilingual country and many Indians especially youths are comfortable with Hindi and English, in addition to their local languages. Hence, they often use more than one language to post their comments on social media. Texts containing more than one language are called "code-mixed texts" and are a good source of input for LI. Languages in these texts may be mixed at sentence level, word level or even at sub-word level. LI at word level is a sequence labeling problem where each and every word in a sentence is tagged with one of the languages in the predefined set of languages. For many NLP applications, using code-mixed texts, the first but very crucial preprocessing step will be identifying the languages in a given text. In order to address word level LI in code-mixed Kannada-English (Kn-En) texts, this work presents i) the construction of code-mixed Kn-En dataset called CoLI-Kenglish dataset, ii) code-mixed Kn-En embedding and iii) learning models using Machine Learning (ML), Deep Learning (DL) and Transfer Learning (TL) approaches. Code-mixed Kn-En texts are extracted from Kannada YouTube video comments to construct CoLI-Kenglish dataset and code-mixed Kn-En embedding. The words in CoLI-Kenglish dataset are grouped into six major categories, namely, "Kannada", "English", "Mixed-language", "Name", "Location" and "Other". Code-mixed embeddings are used as features by the learning models and are created for each word, by merging the word vectors with sub-words vectors of all the sub-words in each word and character vectors of all the characters in each word. The learning models, namely, CoLI-vectors and CoLI-ngrams based on ML, CoLI-BiLSTM based on DL and CoLI-ULMFiT based on TL approaches are built and evaluated using CoLI-Kenglish dataset. The performances of the learning models illustrated, the superiority of CoLI-ngrams model, compared to other models with a macro average F1-score of 0.64. However, the results of all the learning models were quite competitive with each other.

Keywords: Language Identification; Code-mixed texts; Machine Learning; Deep Learning; Transfer Learning

1 Introduction

The measure of mineable information is increasing quickly with the rapid growth of social media. In a country like India where multilingualism is popular, people are comfortable in using more than one language and hence usually use a combination of two or more languages to post their comments or messages on social media. However, these comments may be using single script or multiple scripts. The combination of two or more languages in any text is called code-mixing and is gaining popularity among younger generations mainly to use on social media. English is considered as one of the languages for communication in many countries and the keyboard layout of computers and smartphones by default is of Roman script. Even though there are many apps which can be used to write the text in local languages, however, due to technological glitches most of the users prefer Roman script to write the comments in local or code-mixing languages. Analysis of code-mixed text defines a new research trend due to many challenges. As social media content is not governed by the syntax of any of the languages, short sentences are quite common in addition to incomplete sentences and even words. Words may have a high level of typographical errors intentionally holding creative spellings (gr8 for 'great'), phonetic typescript, word play (goooood for 'good'), and abbreviations (OMG for 'Oh my God!'). Generally, the non-English speakers use English words/sentences (through code-mixing and Anglicism) instead of composing online media text using unicode in their languages. They frequently mix multiple languages in comments/messages to express their thoughts on social media making the analysis of code-mixed text an extremely challenging task.

The preliminary step in analyzing code-mixed texts for various applications is identifying the languages used in these texts efficiently as accuracy of the applications depend on the proper identification of languages. Languages may be mixed at paragraph level, sentence level, word level, or even within a word. Despite a lot of work being done in LI, the problem of LI in code-mixed scenario is still a long way from being illuminated [1]. A code-mixed scenario where words of one language are transcribed with words of other languages as prefix or suffix has lot more troubles, particularly due to conflicting phonetics. In such case, proper context can help in tackling issues like ambiguity. However, capturing context in such data is extremely hard. Furthermore, LI faces the problem of accessible code-mixed dataset to build and evaluate the learning models. The bottleneck of data crisis affects the performance of systems quite a lot, generally because of the issue of over-fitting.

India being a multilingual country has a rich heritage of languages and Kannada is one of the Dravidian languages as well as the official language of Karnataka state. People of Karnataka read, write and speak Kannada but many find it difficult to use Kannada script to post messages or comments on social media. While, technological limitations like keyboards of computers and smartphones is one

reason, another reason may be the complexity of framing words with consonant conjuncts (vattakshara in Kannada). Hence, most of them use only Roman script or a combination of both Kannada and Roman script to post comments on social media. Kn-En code-mixed text on social media is increasing rapidly. Identifying the language of the words in code-mixed social media text is not only interesting but also challenging. LI at word level, is a sequence labeling problem where each and every word in a sentence is tagged with one of the languages in the predefined set of languages. Sequence labeling problem is a special case of Text Classification (TC). Based on ML, DL and TL, this paper explores Learning Approaches for Code-mixed LI (LA-CoLI) at word level for code-mixed Kn-En text. This study includes:

- Developing learning models, namely, CoLI-vectors and CoLI-ngrams based on ML, CoLI-BiLSTM based on DL and CoLI-ULMFiT based on TL approaches
- Developing a code-mixed Kn-En annotated dataset for LI task at word level called as CoLI-Kenglish
- Creating code-mixed Kn-En embeddings for each word by merging word, sub-words and char vectors to build a Skipgram[1] model which will be used as features in learning models to determine the efficiency of combination of vectors in ML and DL approaches
- Training a general domain Language Model (LM) using raw code-mixed Kn-En texts for ULMFiT model

Comments in Kannada YouTube videos are used to create code-mixed Kn-En annotated dataset, code-mixed Kn-En word embeddings and train the LM. Kn-En annotated dataset and Kn-En word embeddings which will be released on request for research purpose. Overall results illustrate the competitive performance among the learning approaches.

2 Related Work

In the ongoing history, a lot of works have been explored on code-mixed data of various language pairs for various applications such as LI, Part-of-Speech (POS) tagging etc. Soumil et al. [1] introduced a novel design for LI of code-mixed Bengali-English (Bn-En) and Hindi-English (Hi-En) data using context information. Their dataset consists of 6000 instances each selected from the datasets prepared by Mandal et al. [2] and Patra et al. [3] for Bn-En and Hi-En

[1] https://towardsdatascience.com/skip-gram-nlp-context-words-prediction-algorithm-5bbf34f84e0c

language pairs respectively. They performed multichannel neural associations merging CNN and LSTM coupled with BiLSTM-CRF for word-level LI of code-mixed data to achieve 93.28% and 93.32% accuracies on the test sets of two language pairs. A novel strategy for incremental POS tagging of code-mixed Spanish/English corpus is proposed by Paul et al. [4]. Utilizing dynamic model switching to get an indicator function which emits term-by-term LI tags, their baseline framework obtained an overall accuracy of 77.27%. The indicator function also regulates the output and picks the most reasonable tagging model to use for a given term. Nguyen et al. [5] introduced experiments on LI of individual words in multilingual conversational data crawled from one of the biggest online networks in Netherlands for Turkish-Dutch speakers during May 2006 to October 2012. Albeit Dutch and Turkish language words rule the discussion, English fixed phrases (e.g. 'no comment', 'come on') are incidentally observed. They evaluated strategies from different points of view on how language recognizable proof at word level can be utilized to analyze multilingual data. The highly informal spelling in online conversations and the events of named substances was used as test set. For their experiments with multilingual online conversations, they first tag the language of individual words utilizing language models and dictionaries and then incorporate context to improve the performance and achieved an accuracy of 98%. Results uncover that language models are more robust than dictionaries and adding context improves the performance.

Sarkar et al. [6] proposed a Hidden Markov Model (HMM) dependent POS tagger for code-mixed Bengali-English (Bn-En), Hindi-English (Hn-En) and Tamil-English (Ta-En) shared task datasets of ICON 2015[2]. They used information from dictionary based methodologies and some word level features to additionally improve the observation probabilities for prediction. Their framework obtained an average overall accuracy (averaged over all three language sets) of 75.60% in constrained mode and 70.65% in unconstrained mode. Yashvardhan et al. [7] presents the methodologies to classify Dravidian code-mixed comments according to their polarity in the evaluation of the track 'Sentiment Analysis for Dravidian Languages in Code-Mixed Text' organized by the Forum of Information Retrieval Evaluation (FIRE) 2020[3]. They trained, validated, and tested the model using the Tamil [8] and Malayalam [9] code-mixed datasets provided by the organizers. Tamil code-mixed dataset consists of 11335 comments for the train set, 1260 for the validation set and 3149 comments for testing the model. Malayalam code-mix dataset consists of 4851 comments for training, 541 for validating, and 1348 for testing the model. Using Long Short-Term Memory (LSTM) network alongside language-explicit pre-processing and sub-word level portrayal to catch the assumption of the content, they obtained F1-scores of 0.61 and 0.60 and overall ranks of 5 and 12 for Tamil and Malayalam datasets respectively.

2 https://ltrc.iiit.ac.in/icon2015/

3 http://fire.irsi.res.in/fire/2020/home

3 Methodology

3.1 Construction of Dataset and Tools

This section describes the functionality used for data collection, preprocessing, training code-mixed word embeddings and building the first ever code-mixed LM for Kn-En language pairs.

3.1.1 Data Collection

Data is the most important part of any study and data for NLP tasks are in form of text and speech. As code-mixed text in Kn-En language pair is required for the proposed work, an efficient module that can scrap data from various sources such as social media platforms, online shopping website, etc. is required. youtube-comment-downloader[4] is modified to download 100000 comments from 373 Kannada YouTube videos which amounts to 72815 sentences after preprocessing. The comments were written only in Kannada or only in English or a combination of Kannada and English and in few cases in other languages namely, Hindi, Telugu and Tamil in addition to Kannada or English or both. However, the script of these comments is either Roman or Kannada or a combination of Roman and Kannada. The workflow of data collection module is shown in Figure 1. Data collection module accepts a list of Kannada YouTube video ids as input, downloads the comments, preprocesses them and provides as output a list of sentences extracted from the comments posted on each video.

3.1.2 Preprocessing

Comments in social media are unstructured, messy, contain incomplete sentences and words in short forms in addition to code-mixing of two or more languages. All these features increase the complexity of analyzing code-mixed text. Hence, the first step in analyzing these texts is preprocessing, which includes removing duplicate comments, comments in Kannada script, short comments (less than 3 words) and comments consisting of only English words, emojis and unprintable characters. After preprocessing, roughly 90% of the data is used as raw data to train Kn-En tokenizer, code-mixed Kn-En word embeddings and code-mixed LM for Kn-En language pairs. Remaining 10% of the data is processed further to create annotated dataset for LI, at the word level. The major problem faced in analyzing code-mixed text is lack of normalization of words.

[4] https://github.com/egbertbouman/youtube-comment-downloader

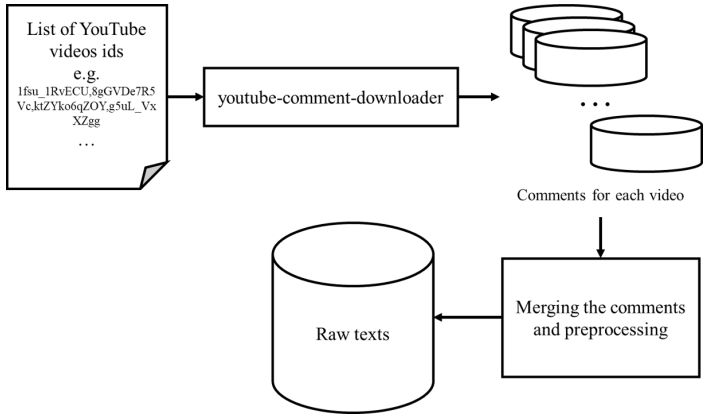

Figure 1
Data Collection Module

3.1.3 Creation of CoLI-Kenglish Dataset

A small portion (10%) of the preprocessed code-mixed texts are selected randomly and tokenized into words. These words are tagged manually by two native Kannada speakers (these people are trained about concepts of code-mixed texts and LI task) to generate CoLI-Kenglish dataset. 19432 unique words extracted from nearly 7000 sentences are categorized into 6 classes namely, 'Kannada', 'English', 'Mixed-language', 'Name', 'Location' and 'Other'. While the first two classes represent Kannada and English words respectively, 'Mixed-language' class represents word created using a combination of Kannada and English in any order. 'Name' class represents the names of persons and 'Location' class the names of locations or places. Any other words are represented as 'Other' class. The words described by 'Mixed-language' pose a real challenge to LI task as these words are framed by various combinations of English/Kannada words and Kannada/English affixes (prefixes and suffices). Beauty and also the complexity of these mixed-language words is that the word pattern depends on an individual and users posting comments on social media is increasing day-by-day. Description and samples of tokens are given in Table 1.

3.1.4 Word Embeddings

Word embeddings are seen as the key ingredient for many NLP tasks and has been proved as an efficient representation for characterizing the statistical properties of natural languages [11]. In addition to providing text to numeric vector conversion that is understandable to Neural Networks (NN), they model the complex characteristics of words, such as syntax and semantics which vary across linguistic contexts. Word embeddings consisting of word, sub-words, and char vectors is

trained on 90% of the preprocessed Kn-En code-mixed raw data which is in the form of sentences. The steps to train the vectors as follows:

- **Word vectors:** By tokenizing sentences to words, code-mixed word2vec model of size 200 is trained on the words based on Skipgram model using gensim[5] library
- **Sub-word vectors:** A sub-word is a substring of a word. BPEmb[6] tools are used to split each word to sub-words. Similar to word vectors a code-mixed sub-word2vec of size 100 is trained on the sub-words based on Skipgram model
- **Character vectors:** A char2vec model of size 30 is trained on all characters which appear in the text based on Skipgram model

The sizes of the vector's dimensions selected for the proposed word embeddings are set based on the average unique tokens of each type in the dataset (words, sub-words, and characters). A sentence is made up of several words and each word can be decomposed into several sub-words and several characters. Hence, a word vector is extended by sub-words vectors and character vectors. In order to have a fixed length vector representation for words, the number of sub-words is fixed as the maximum of the number of sub-words of all the words in the vocabulary and similarly the number of characters is fixed as the maximum of the number of characters of all the words. For each word, word2vec, sub-word2vec, and char2vec Skipgram based models are trained as mentioned above.

As the number of sub-words is not the same for all words, sub-word2vec of a word is padded with zeros depending on the difference between the maximum of the number of sub-words of all the words and the number of sub-words in a word. Similarly, char2vec is padded with zeros depending on the difference between the maximum of the number of characters of the words and number of characters in a word. Finally, word2vec, sub-word2vec, and char2vec vectors are merged together to obtain one vector for each word as shown in Figure 2. Table 2 gives a glimpse of the size of the all vectors used to obtain a vector for a word.

3.1.5 Kn-En Tokenizer

Tokenization is an initial but very crucial step in many token level classification tasks such as POS [10], Named Entity Recognition (NER), and token level LI [14]. Many pre-trained tokenizers are available in NLTK[7] and iNLTK[8] libraries for tokenizing Indian languages but tokenizers for code-mixed text are rarely

5 https://pypi.org/project/gensim/
6 https://nlp.h-its.org/bpemb/
7 https://www.nltk.org/
8 https://pypi.org/project/inltk/

found. SentencePiece[9] is an unsupervised text tokenizer that utilizes sub-words units e.g., Byte-Pair-Encoding (BPE) [15] and unigrams [16] with the extension of directly training from raw sentences. A Kn-En code-mixed tokenizer is trained on 90% of the preprocessed Kn-En code-mixed raw texts with a vocabulary size of 10000 using SentencePiece tools. Figure 3 illustrates the procedure of training Kn-En tokenizer and generating vocabulary.

Table 1
Description and samples of tokens in CoLI-Kenglish dataset

Category	**Description**	**Samples**
Kannada	Kannada words written in Roman script	kopista (one who get angry soon), baruthe (will come), barbeku (must come)
English	Pure English words	small, need, take, important
Mixed-language	Combination of Kannada and English words in Roman script	coolagiru (cool + agiru, be cool), leaderge (leader + ge, to a leader), homealli (home + alli, inside home)
Name	Words that indicate name of person (including Indian names)	Madhuswamy, Hemavati, Swamy
Location	Words that indicate locations	Karnataka, Tumkur, Bangalore
Other	Words not belonging to any of the above categories and words of other languages	Znjdjfjbj – not a word ನದನ – kannada word in kannada script उसके – hindi word in Devanagari script uske – hindi word in Roman script நான் – tamil word in Tamil script

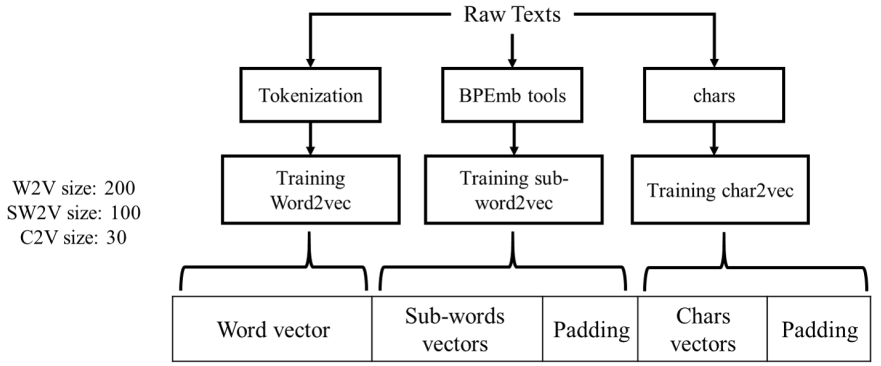

Figure 2
Merging word2vec, sub-word2vec, and char2vec vectors

[9] https://github.com/google/sentencepiece

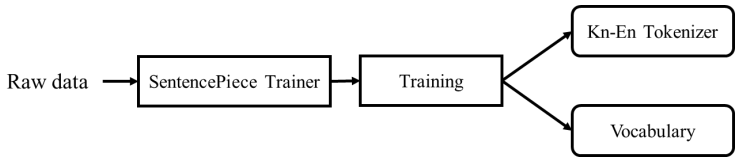

Figure 3
Procedure of generating Kn-En tokenizer and Vocabulary

3.1.6 Language Model

LM is a probability distribution over the sequence of words, in other words, LM is able to predict next word(s) in a given sequence of words and window [13]. It has applications in NLP tasks such as "Smart Compose" feature in Gmail that suggests next words in sequence. Voice to text conversion, speech recognition, sentiment analysis, text summarization, and spell correction are other NLP tasks where LMs can be used. Further, a LM can be seen as a statistical tool that can learn and analyze the natural languages' patterns. LM has got more attention with TL where the knowledge of one (source) model is transferred to another (target) model. Raw text collected from YouTube video comments (as mentioned in section 3.1.1) have been used to train a tokenizer using SentencePiece library as explained in section 3.1.5. This is then used along with raw texts to train the LM for Kn-En code-mixed text with a vocabulary size of 10000. Fast.ai[10] library is used to train the LM for 150 training epochs with various learning rates. More details are given in 3.2.4.

3.1.7 N-grams Model

One of the challenges of LI is the structure of words in natural language. For example, it is very common in English to see letter "q" to be followed by letter 'u' in words such as question, quarrel, qualifications, quietness, etc. However, this rule is not followed in many code-mixed texts. Since one of the primary advantages of character n-grams is language independence [17] it can be utilized for any language including code-mixed texts to capture the structure of words that has been written in a different script. In this study, a feature engineering module that generates a feature set for a given text is implemented. The feature set comprises of prefixes and suffixes of length 1, 2 and 3 along with char ngrams (n = 2, 3, 5) from words, and char ngrams (n = 1, 2, 3) from sub-words.

[10] https://nlp.fast.ai/

3.2 Learning Models

Four learning models, namely, CoLI-ngrams, CoLI-vectors, CoLI-BiLSTM, and CoLI-ULMFiT are proposed for the Kn-En code-mixed LI task at word level. The learning models based on ML, DL, and TL approaches are constructed and evaluated using CoLI-Kenglish dataset and the tools constructed as mentioned above. All the four learning models are explained below:

Table 2

Glimpse of the all vectors size all to form a vector for a word

Size of word2vec = 200
Size of sub-word2vec = 100
Size of char2vec = 30
Maximum of the number of sub-words of all words in the vocabulary = 8
Maximum of the number of characters of all the words = 10
Total size of word2vec = size of word2vec + 8 x size of sub-word2vec + 10 size of char2vec = 200 + 8x100 + 10x30 = 1300
If a word 'w' has 5 sub-words and 6 characters, then word vector will be a combination of word2vec + 5 sub-word2vec + (8-5) sub-word2vec zero paddings + 6 char2vec + (10-6) char2vec zero paddings
Sub-word2vec zero paddings will be of the size of sub-word2vec and char2vec zero paddings will be of the size of char2vec.

3.2.1 CoLI-Ngrams

This model is an ensemble of three ML classifiers namely, Linear SVC (LSVC), Multi-Layer Perceptron (MLP) and Logistic Regression (LR) with 'soft' voting. Values of the parameters used in these classifiers are given in Table 3. Figure 4 presents the structure of CoLI-ngrams model which is fed with count vectors of ngrams obtained from a feature engineering module described in section 3.1.7.

Char ngrams from sub-words are extracted in two steps: i) extracting sub-words from words using BPEmb and ii) generating char ngrams for extracted sub-words. BPEmb provides pre-trained sub-words embeddings for 275 languages that are trained on texts from Wikipedia [18]. An embedding with a vocabulary size of 10000 is downloaded for English language to encode and extract sub-words from code-mixing text which helps to extract exact English words from code-mixed words. In code-mixed words, one part of the word may be an English word and rest can be Kannada suffix or prefix or with some characters which do not have any meaning in any language. In other words, sub-Words help in the generation of words that are rarely been seen in training set. Table 4 illustrates the samples of words and corresponding features generated for CoLI-ngrams.

Table 3
Parameters for estimators in CoLI-ngrams and CoLI-vectors

Estimators	Parameters
Linear SVC	kernel='linear',probability=True
MLP	hidden_layer_sizes=(150,100,50), max_iter=300, activation = 'relu', solver='adam', random_state=1
LR	Default parameters

3.2.2 CoLI-Vectors

This model uses estimators as in CoLI-ngrams model but trained on vectors for words in the training set by utilizing embedding module that generates word embed-dings for words, sub-words and characters from raw text as discussed earlier. The purpose of developing CoLI-vectors model is to compare the performances of voting classifiers with different features and also to compare the efficiency of proposed word embedding architecture using ML and DL approaches. Figure 5 gives the structure of CoLI-vectors model.

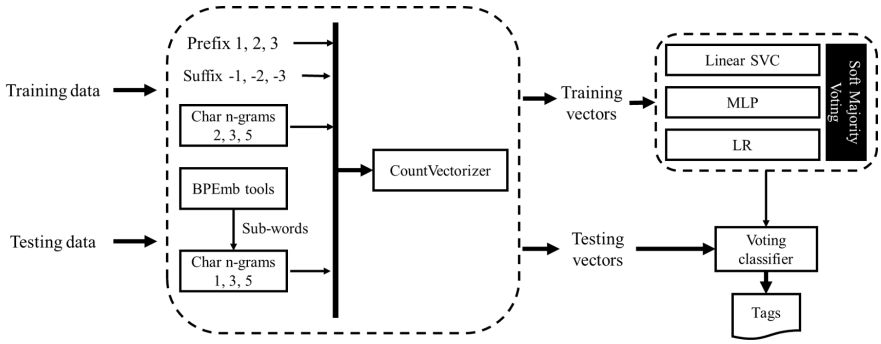

Figure 4
Structure of CoLI-ngrams model

Table 4
Samples of words and corresponding features generated for CoLI-ngrams model

Word	Language (tag)	Sub-words and ngrams of sub-words
Nayigalige (in English: for dogs)	Kannada (Kn)	'nayigalige', 'ige', 'nay', 'ge', 'na', 'e', 'n', '_nay', 'nayi', 'ayig', 'yiga', 'igal', 'gali', 'alig', 'lige', 'ige_','_ay_', '_ig_', '_al_', '_ig_'
Dogsgalige (in English: for dogs)	Mixed-language (Kn-En)	'dogsgalige', 'ige', 'dog','ge', 'do', 'e', 'd', '_dog', 'dogs', 'ogsg', 'gsga', 'sgal', 'gali', 'alig', 'lige', 'ige_', '_og_', '_al_', '_ig_'

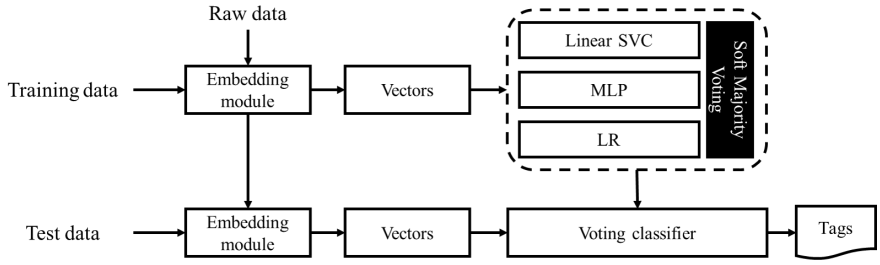

Figure 5
Structure of CoLI-vectors model

3.2.3 CoLI-BiLSTM

Learning models based on DL approach have excelled conventional models based on ML approach in various NLP tasks, such as Sentiment Analysis, NER etc. [19]. CoLI-BiLSTM model is a sequence processing model based on Bidirectional Long Short Term Memory (BiLSTM) architecture. It utilizes the feature vectors obtained from proposed word embedding model. A BiLSTM comprises of two LSTMs that take the input in forward as well as in backward direction. In other words, at every time step BiLSTM networks have both backward and forward information about the sequence [20-21]. CoLI-BiLSTM model consists of layers summarized in Table 5. It includes input and embedding layers to load training data and weights from word embedding model and a BiLSTM layer followed by time_distributed layer. The purpose of using time_distributed layer is to keep one-to-one relations on input and output on RNNs including LSTM and BiLSTM. This scenario is commonly used in NNs in sequence classification tasks such as POS, NER, etc. The structure of CoLI-BiLSTM model is shown in Figure 6.

3.2.4 CoLI-ULMFiT

The approach of transferring knowledge of one model called source model to improve the performance of the other model called target model is called TL. Universal Language Model Fine-Tuning (ULMFiT) is one of architectures that utilize the concept of TL [12]. It consists of training a LM and then transferring the obtained knowledge and fine-tuning the target model with the dataset provided for the given task. Usually in NLP tasks the data used for training LM will be a large corpus with same or different domain from the dataset used for target task. The benefit of training a LM is that once a pre-trained LM is ready, its knowledge can be utilized in different NLP tasks including token level or text level classification, summarization, etc. A pre-trained LM understands the general features of language and then fine-tuning the LM using target task dataset helps in obtaining more properties of specific task. Following the ULMFiT architecture adopted from [13], CoLI-ULMFiT model includes training a LM from

preprocessed code-mixed texts (section 3.1.2), transferring and then fine-tuning the weights using training set (section 3.1.3- CoLI-Kenglish) and finally using weights and knowledge obtained from LM in target LI model. Figure 7 presents the overview of CoLI-ULMFiT model.

Fast.ai library provides necessary modules for the implementation of ULMFiT model. text.models tools from Fast.ai library is used to construct both LM and target LI models. An encoder for Average-Stochastic Gradient Descent (SGD) Weight-Dropped LSTM (AWD-LSTM) implemented using text.models tools consists of a word embedding of size 400, 3 hidden layers and 1150 hidden activations per layer-plugged in with a decoder plus classification layers to create a TC [22].

4 Experiments and Results

4.1 Datasets

Inspired by [23-25] in utilization of YouTube code-mixed comments, CoLI-Kenglish dataset has been developed. The construction of the CoLI-Kenglish dataset for LI at word level is mentioned in Section 3.1.3 and the distribution of labels in CoLI-Kenglish dataset is shown in Figure 8. Statistics of raw data and CoLI-Kenglish dataset is summarized in Table 6. Since texts in social media generally do not follow any rules the tagged dataset is highly imbalanced which may result in less F1 score. The dataset also illustrates that nearly 44.8% words are Kannada words, about 7.5% words are Kn-En mixed language words like “Dogsgalige” (meaning ‘for dogs’, dogs is an English word and ‘galige’ is a suffix in Kannada) and about 32.32% words are English words. Approximately, 70% of the tagged dataset is used for training and remaining 30% for testing.

Table 5
Layers in CoLI-BiLSTM

Layer (type)	**Output shape**	**Param #**
Input layer	[(none, 1000)]	0
Embedding layer	(none, 1000, 1000)	19162000
BiLSTM	(none, 1000, 600)	3122400
time_distributed layer	(none, 1000, 7)	4207

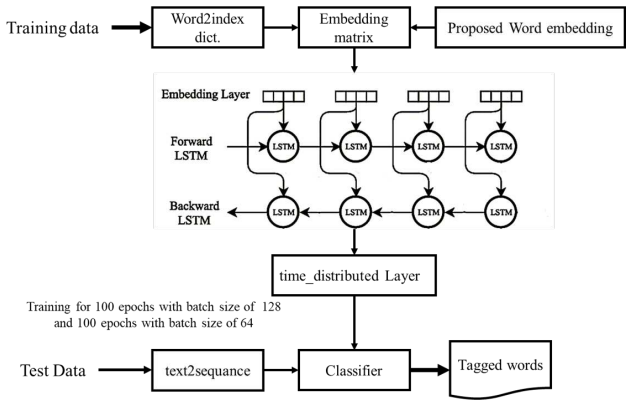

Figure 6
Structure of CoLI-BiLSTM model

4.2 Results

This study provides a comparison of the performances of the proposed models for word level LI task in Kn-En code-mixed texts and the results are shown in terms of macro average metrics. Kannada and English are two completely different languages in various terms such as grammar, script, structure, etc., but still performances of models are promising considering the noisiness of the data. However, it is expected in closely related languages, e.g. English-German or Spanish-Italian mixed texts, LI task will be more challenging but availability of more tools for such languages enable models to have more efficient performances.

The performances of the proposed learning models in addition to the performances of individual estimators in case ML models are shown in Table 7. CoLI-Kenglish dataset for word level LI consists of 6 categories and category-wise results in terms of Precision, Recall and F1-score of all the proposed models are shown in Table 8. Further, category-wise comparison of macro average F1-scores of the proposed models is illustrated in Table 9. Results of ML models illustrate that ML classifiers (both individual and ensembled) with character ngrams and affixes outperformed the ML classifiers with proposed word embeddings.

Table 6
Statistics of datasets

Dataset	Type	No. sentences	No. words
Raw texts	unannotated	72135	594680
CoLI-Kenglish DS	annotated	700	19432

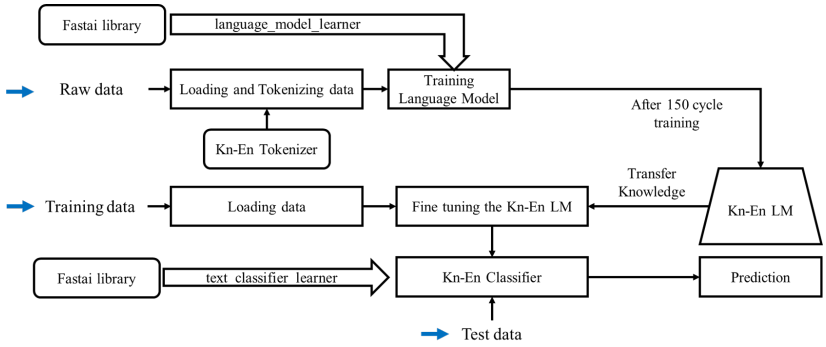

Figure 7
Overview of CoLI-ULMFiT model

CoLI-BiLSTM model has been trained for 200 epochs (100 epochs with batch size of 128 and 100 epochs with batch size of 64. The results illustrate that that CoLI-ngrams model based on ML approach trained on a subset of morphological features including char ngrams from words and sub-words along with affixes beats the other models. CoLI-ULMFiT model has obtained relatively good performance except for words belonging to "location" class which is due to lack of sufficient samples in tagged dataset. Training an LM for ULMFiT architecture efficiently requires very huge dataset.

Table 7
Comparison of performances among ML models and individual estimators

Classifier	**Features**	**Performance**		
		Precision	**Recall**	**F1-score**
Linear SVC	word embeddings	0.35	0.60	0.37
LR	word embeddings	0.37	0.69	0.40
MLP	word embeddings	0.37	0.64	0.39
CoLI-vectors	word embeddings	0.36	0.69	0.39
Linear SVC	ngrams + affixes	0.73	0.57	0.62
LR	ngrams + affixes	0.74	0.55	0.60
MLP	ngrams + affixes	0.70	0.60	0.63
CoLI- ngrams	**ngrams + affixes**	**0.73**	**0.60**	**0.64**
CoLI-BiLSTM	Proposed vectors	0.61	0.74	0.63
CoLI-ULMFiT	A code-mixed LM	0.42	0.42	0.41

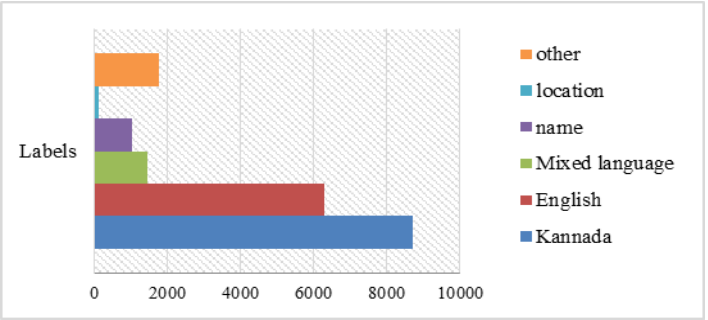

Figure 8
Labels distribution over the CoLI-Kenglish DS

Conclusions

This study explores four learning models, the CoLI-ngrams, the CoLI-vectors, the CoLI-BiLSTM and the CoLI-ULMFiT for Kn-En code-mixed LI at the word level. While CoLI-ngrams and CoLI-vectors are based on ML approaches, CoLI-BiLSTM and CoLI-ULMFiT are based on DL and TL approaches respectively. Due to lack of Kn-En code-mixed dataset at word level for LI and tools to process Kn-En code-mixed data, comments in Kannada YouTube videos were scrapped and processed to construct CoLI-Kenglish tagged dataset, Kn-En code-mixed word embeddings and Kn-En code-mixed LM. CoLI-Kenglish dataset was manually tagged by Kannada speakers and grouped into six categories. Kn-En code-mixed word embeddings was constructed by merging word, sub-words, and characters vectors which were built using Skipgram model. This embedding is used as features in CoLI-vectors and CoLI-BiLSTM models and a subset of morphological characteristics are used as features in CoLI-ngrams model.

CoLI-ULMFiT utilizes ULMFiT architecture to transfer the knowledge of a pre-trained Kn-En LM to a LI model. The results of the proposed models illustrate that CoLI-ngrams utilizing morphological features outperformed all other models with an average macro F1-score of 0.64. Further, CoLI-ULMFiT model also obtained similar overall performance, except for the “location” category. The results obtained by CoLI-vectors and CoLI-BiLSTM models illustrate the superiority of DL approach over ML approach in using proposed embedding.

As the generated Kn-En code-mixed LM and Kn-En code-mixed word embedding can be used for other Ka-En code-mixed NLP tasks they will be released publicly along with CoLI-Kenglish dataset. In the future, it is planned to enrich both unannotated and annotated dataset, construct a balanced label distribution and to explore different feature sets and models based on different learning approaches. It is also planned, to bring morphological features into the vector space, to determine the enhancement potential of the proposed models.

Table 8
Category-wise results of the proposed models

Model	Metric	Labels					
		English	**Kannada**	**Mixed-language**	**Name**	**location**	**other**
CoLI-vectors	Precision	0.66	0.94	0.02	0.29	0.13	0.14
	Recall	0.87	0.60	0.77	0.56	0.86	0.50
	F1-score	0.75	0.74	0.05	0.38	0.22	0.22
CoLI-ngrams	Precision	0.83	0.82	0.87	0.56	0.75	0.57
	Recall	0.87	0.89	0.66	0.45	0.27	0.44
	F1-score	**0.85**	**0.85**	0.75	0.50	0.39	0.50
CoLI-BiLSTM	Precision	0.74	0.87	0.69	0.21	0.14	1.00
	Recall	0.75	0.70	0.87	0.40	0.71	1.00
	F1-score	0.74	0.78	**0.77**	0.27	0.24	1.00
CoLI-ULMFiT	Precision	0.68	0.71	0.68	0.09	0.0	0.34
	Recall	0.81	0.71	0.67	0.03	0.0	0.30
	F1-score	0.74	0.71	0.67	0.04	0.0	0.32

Table 9
Category-wise comparison of F1-score of the proposed models

Model	English	Kannada	Mixed-language	Name	location	other
CoLI-vectors	0.75	0.74	0.05	0.38	0.22	0.22
CoLI- ngrams	0.85	0.85	0.75	0.5	0.39	0.5
CoLI-BiLSTM	0.74	0.78	0.77	0.27	0.24	1
CoLI-ULMFiT	0.74	0.71	0.67	0.04	0	0.32

Acknowledgments

The work was supported by the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico and grants 20211784, 20211884, and 20211178 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico.

References

[1] Mandal, Soumil, and Anil Kumar Singh. "Language Identification in Code-Mixed Data using Multichannel Neural Networks and Context Capture". arXiv preprint arXiv:1808.07118, 2018

[2] Mandal, Soumil, Sainik Kumar Mahata, and Dipankar Das. "Preparing Bengali-English Code-mixed Corpus for Sentiment Analysis of Indian Languages". arXiv preprint arXiv:1803.04000, 2018

[3] Patra, Braja Gopal, Dipankar Das, and Amitava Das. "Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task@ ICON-2017". arXiv preprint arXiv:1803.06745, 2018

[4] Rodrigues, Paul, and Sandra Kübler. "Part of Speech Tagging Bilingual Speech Transcripts with Intrasentential Model Switching". In AAAI Spring Symposium: Analyzing Microtext. 2013

[5] Nguyen, Dong, and A. Seza Doğruöz. "Word Level Language Identification in Online Multilingual Communication". In Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 857-862, 2013

[6] Sarkar, Kamal. "Part-of-speech Tagging for Code-mixed Indian Social Media text at ICON 2015". arXiv preprint arXiv:1601.01195, 2016

[7] Sharma, Y., & Mandalam, A. V. (2020) Bits2020@ Dravidian-CodeMix-FIRE2020: Sub-Word Level Sentiment Analysis of Dravidian Code Mixed Data. In FIRE (Working Notes), pp. 503-509, 2020

[8] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus creation for sentiment analysis in code-mixed Tamil-English text, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL),European Language Resources association, Marseille, France, 2020, pp. 202-210

[9] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A sentiment analysis dataset for code-mixed Malayalam-English, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced lan-guages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources associa-tion, Marseille, France, 2020, pp. 177-184

[10] Wang, Peilu, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. "Part-Of-Speech Tagging With Bidirectional Long Short-Term Memory Recurrent Neural Network". ArXiv preprint arXiv: 1510.06168, 2015

[11] Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. "Deep Contextualized Word Representations". ArXiv preprint arXiv: 1802.05365, 2018

[12] B. Fazlourrahman, and H. L. Shashirekha. "Puner-Parsi ULMFiT for named-entity recognition in Persian texts." Congress on Intelligent Systems, pp. 75-88, Springer, Singapore, 2020

[13] Howard Jeremy, and Sebastian Ruder. "Universal Language Model Fine-Tuning for Text Classification". ArXiv preprint arXiv: 1801.06146, 2018

[14] Rai A. and Borah, S. "Study of Various Methods for Tokenization". In Applications of In-ternet of Things. Springer, Singapore, pp. 193-200, 2020

[15] S Rico, H Barry, and B Alexandra."Neural Machine Translation of Rare Words with Sub-word Units". Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL, Germany, pp. 1715-1725, 2016

[16] Taku Kudo, "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates". ArXiv preprint arXiv: 1804.10959, 2018

[17] Kruczek, J., Kruczek, P. and Kuta, M., June. "Are n-gram Categories Helpful in Text Classification?" In International Conference on Computational Science, Springer, Cham. pp. 524-537, 2020

[18] Balouchzahi, F., & Shashirekha, H. L. (2020) MUCS@ Dravidian-CodeMix-FIRE2020: SACO-SentimentsAnalysis for CodeMix Text. In FIRE (Working Notes) pp. 495-502, 2020

[19] Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J. "Deep Learning based Text Classification: A Comprehensive Review". arXiv preprint arXiv:2004.03705, 2020 Apr 6

[20] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF Models for Sequence Tagging". ArXiv preprint arXiv: 1508.01991, 2015

[21] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory". Neural Computation 9, No. 8, pp. 1735-1780, 1997

[22] Stephen Merity, Nitish Shirish Keskar, and RichardSocher. 2017, "Regularizing and Optimizing LSTMLanguage Models".arXiv preprint arXiv: 1708.02182

[23] A. Hande, R. Priyadharshini, B. R. Chakravarthi, "KanCMD: Kannada Code-Mixed Dataset for Sentiment Analysis and Offensive Language Detection", in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, Association for Computational Linguis-tics, Barcelona, Spain (Online), 2020, pp. 54-63

[24] B. R. Chakravarthi, HopeEDI: "A Multilingual Hope Speech Detection Dataset for Equality, Diversity, and Inclusion", in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, 2020

[25] B. R. Chakravarthi, V. Muralidaran, "Findings of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion", in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, Association for Computational Linguistics, Kyiv, 2021, pp. 61-72, URL: https://aclanthology.org/2021.ltedi-1.8

Automatic Abusive Language Detection in Urdu Tweets

Maaz Amjad[1], Noman Ashraf[1], Grigori Sidorov[1], Alisa Zhila[2], Liliana Chanona-Hernandez[3], Alexander Gelbukh[1][1]

[1]Instituto Politécnico Nacional, Centro de Investigación en Computación (CIC), Gustavo A. Madero, 07738 Mexico City, Mexico

[2]Independent Researcher, San Francisco, CA 94103, USA

[3]Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME), Gustavo A. Madero, 07340, Mexico City, Mexico

e-mails: maazamjad@phystech.edu, noman@nlp.cic.ipn.mx, Sidorov@cic.ipn.mx, alisa.zhila@ronininstitute.org, lchanonah2100tmp@alumnoguinda.mx, gelbukh@cic.ipn.mx

Abstract: Abusive language detection is an essential task in our modern times. Multiple studies have reported this task, in various languages, because it is essential to validate methods in many different languages. In this paper, we address the automatic detection of abusive language for tweets in the Urdu language. The study introduces the first dataset of tweets in the Urdu language, annotated for offensive expressions and evaluates it by comparing several machine learning methods. The Twitter dataset contains 3,500 tweets, all manually annotated by human experts. This research uses three text representation techniques: two count-based feature vectors and the pre-trained fastText word embeddings. The count-based features contain the character and word n-gram, while the pre-trained fastText model comprises word embeddings extracted from the Urdu tweets dataset. Moreover, this study uses four non-neural network models (SVM, LR, RF, AdaBoost) and two neural networks (CNN, LSTM). The study finding reveals that SVM outperforms other classifiers and obtains the best results for any text representation. Character tri-grams perform well with SVM and get an 82.68% of F1 score. The best-performing words n-grams are unigrams with SVM, which obtain 81.85% F1 score. The fastText word embeddings-based representation yields insignificant results.

Keywords: Twitter corpus; Abusive language detection; Urdu language; Machine learning

[1] Corresponding author: Alexander Gelbukh

1 Introduction

Abusive language detection is an alluring concept. People use language to highlight, depict, elicit, instruct, and urge to inform the nuances of themselves and their worlds [1], some use it for a good cause, and some use it for spite. Impacts of abusive language are detrimental, ranging from short-term emotional reactions (outrage, dread, self-fault, etc.) to long-term mental health effects (low confidence, misery, etc.), causing psychological and medical problems (rest issues, migraine, dietary issues, etc.) [2] [3]. According to the Guardian report[2], abusive language can change human behavior. Although several prevention and intervention strategies were introduced, usage of abusive language on social media increased in recent years.

The task of abusive language detection is widely investigated in languages other than the English language [5-10]. Some studies discussed linguistic aspects and linguistic resources in different languages, such as Arabic [6], German [9], Japanese [10], Indonesian [7], Danish [8], and Portuguese [5]. Although automatic abusive language detection is still in its earliest stage, no study to date investigated abusive language detection with automatic manners on Twitter in Urdu, a local language of Pakistan, having over 230 million worldwide native speakers[3]. In addition, Urdu is viewed as one of the best ten most spoken languages on the planet. According to the point of view of NLP tools, inaccessibility, and the shortage of annotated data [4], Urdu is viewed as a low-resource language. Therefore, the study mainly focuses on counting features (N-gram) and word embeddings as feature vectors for abusive language identification tasks using Urdu tweets.

Abusive language detection is a challenging task. Recently social networks established themselves as the primary platforms for discussion, sharing ideas, and emotions. Being free and accessible, they lack language moderation. While most of the users stay cordial and polite, some occasionally express themselves in a manner that is obscene/profane and even might be rude or offensive to other users. The profane and objectionable content on social media might severely affect the addressee's emotional state and deteriorate life quality. Therefore, automatic abusive language identification is an invaluable measure. Among all, it can blow with a shovel the obscenities or indecent content for increased child protection.

Twitter is recognized as a social network where users can only post short text posts. To mitigate the use of abusive language on its platform, Twitter characterized harmful conduct as an endeavor to molest, threaten, or quietness of another person's voice[4]. Abuse is characterized by physical or psychological maltreatment. For example, I love you, but you are chubby. In this context, the word chubby might be perceived as diminishing one's appearance. Therefore, such reference to human appearance, personality or behavior might be considered a vile aspersion.

2 https://www.theguardian.com/education/2011/oct/03/researchdemonstrates-language-affects-behaviour

3 https://www.statista.com/statistics/266808/the-most-spokenlanguages-worldwide/

4 https://help.twitter.com/en/rules-and-policies/abusive-behavior

Till today, no work identified abusive language using Urdu tweets, and no related corpora were recently gathered to the best of our knowledge. Moreover, no relevant Twitter dataset containing Urdu tweets was recently compiled. Therefore, this study presents the first balanced dataset containing abusive and non-abusive tweets in Urdu to address the automatic detection of abusive tweets.

Furthermore, as Urdu stays a relatively low resource language, we explore how data-intensive approaches such as neural networks and embedding-based text representation perform compared to count-based features and linear classifiers.

This study makes three main contributions, discussed as follows:

- Presents the first dataset in the Urdu language, for the automatic detection of abusive language using Twitter postings in Urdu, manually labeled by experts using given guidelines. This study also clarifies the dataset collection and annotation process that addresses the task of automatic abusive language detection, in Urdu.
- Baseline results utilize five non-neural network models (RF, Ada-Boost, MLP, LR, SVM) and two neural network models (LSTM, 1D-CNN). Three text representations techniques are used: two count-based and the pre-trained fastText word embeddings to identify abusive postings on the Twitter dataset.
- Analyzes the performance of different machine learning and deep learning algorithms on the proposed dataset.

The remaining paper is divided into different sections: the recent studies on the identification of abusive language are highlighted in Section 2. Section 3 examines the guidelines used to create and annotate the dataset. The results obtained in the experimental setup using machine learning (ML) and deep learning (DL) algorithms are presented in Section 4. Section 5 analyzes the performance of various classifiers and explains the results in detail. Finally, in Section 6, this study provides conclusions to our work.

2 Literature Review

This section first discusses the definition of abusive language and subsequently sheds light on existing research in the automatic detection of abusive language.

2.1 Defining and Characterizing Abusive Language

Twitter is characterized as one of the top five social networks, where a substantial number of users experience unethical communication and bullying. Using this platform, users can access a large active Twitter community (more than 330 million)

and write a tweet with a maximum of 280-characters[5]. Several recent studies [11-13] highlighted that abusive language, and bullying cases are often reported on Twitter that contain injurious consequences for active tweeter users. Thus, Twitter took some safety measures and described some policies to control the usage of abusive language on its platform. According to the new guidelines, messages from obscure clients who have no profile picture will be removed. If the tweet is detected containing abusive words, this will lead to removing the user account from Twitter[6]. Nonetheless, Twitter must take more robust steps, especially distinguishing harmful tweets in different dialects since individuals utilize different abusive words in other languages.

2.2 Available Methods for Abusive Language Detection

Twitter permits its users to create new profiles, follow existing profiles, send messages and tags (both private and public), uploading status, images and videos. Within minutes and seconds, a single tweet can target a massive number of audiences, primarily through commenting, liking, sharing, and re-tweeting mechanisms. Among all the social media platforms, ordinary people widely use Twitter, yet this social network also attracted politicians, government organizations, and a government media spokesperson to release government statements (i.e., policies). Eventually, it creates a space for ignorant users to spread humiliating, hostile, and infancy comments with high velocity to a broader range of people.

Online platforms started introducing new policies to counterfeiting this issue because young people were a target group for bullying victimization. For example, Instagram started to fight against bullies by introducing shadow banning online abusers (i.e., limiting the user (bully) who used abusive language from publishing new posts or commenting on others' posts). Instagram introduced this system to mitigate cyberbullying events[7]. Likewise, another platform called Ask.fm [7] (an online website that permits its users to ask each other questions without disclosing identity) also introduced new policies to avoid discrepancies between users and other threats (life threats).

Numerous studies discussed abusive language detection and proposed various methods ranging from traditional machine learning to neural network-based models. Two features, such as character-level and word-level representations, were used to detect abusive language [14, 15, 19]. Furthermore, another study [18] highlighted that feature engineering, e.g., n-grams and POS tags, are extremely fruitful in machine learning methods. Other studies [14-18] used various traditional machine learning models, such as support vector machines, random forest, decision tree,

5 https://www.statista.com/ statistics/282087/number- ofmonthly- active- twitter- users

6 https://social.techcrunch.com/2017/02/16/twitter-starts-puttingabusers-in-time-out

7 https://qz.com/1661410/instagram-wants-to-fight-bullies-byshadowbanning-them-and-telling-them-they-are-bullies/

logistic regression, and deep learning models like BERT [32] and Roberta [32], to identify hate speech. Further, several neural network architectures [8, 15, 17, 20] were used to identify the abusive language in Twitter posts. Recent studies [20] concluded deep learning models, such as convolutional neural networks (CNN) along with recurrent neural networks (RNN) [8] outperformed traditional ML classifiers, such as Logistic Regression [8] [19] and SVM [14-17]. Obscenity and offensive language detection tasks focused on languages, such as English [5] [8] [14-19] [20-24], Indonesian [7], Arabic [6], Portuguese [5], German [9], Japanese [10] and Danish [8]. Table 1 summarizes the recent works that examined abusive language detection in different languages.

3 Abusive Tweets Dataset in Urdu

This section explains the steps followed for dataset creation and data annotation. The dataset creation is divided into two stages (i) dataset crawling and (ii) dataset annotation.

3.1 Data Crawling

This study used Twitter API[8] to extract the tweets in the Urdu language using abusive keywords. To crawl tweets from Twitter, some keywords are used that contained only either a word or at least two abusive words. A dictionary containing abusive words and phrases in Urdu was manually created based on the most frequent words used on different social media platforms. The complete list of the keywords used to crawl the abusive tweets can be accessed[9].

This study collected the dataset for 20 months, starting from 01 January 2018 to 30th August 2019. This time interval was chosen primarily due to the General Elections in Pakistan held in July 2018. Typically, during or near election season, supporters of different political parties express their emotions and show antagonistic behavior to each other.

According to a recent report, although abusive language and threats to anyone are not confined to politics[10], some people use abusive and threatening language as a potent weapon for a political campaign.

Similarly, some people use social networks to use vulgar language to support a specific political party. For example, the current prime minister of Pakistan claimed that the daughter of the Ex-prime minister compelled her supporters to abuse him

8 https://developer.twitter.com/en/docs/tweets/search/apireference/get-search-tweets

9 https://github.com/MaazAmjad/Abusive_dataset.git

10 https://www.cbc.ca/news/politics/violence-vandalismcampaign-rise-1.6177269

in public[11].

Table 1
Overview of the recent studies to identify the abusive language in different languages

Comparison of the state-of-the-art in abusive language detection				
Language	Platform	Feature extraction method	Classifier	Reference
English	Twitter	Char n-gram (1-4)	LR, Graph Convolutional Network	[20]
English	NewsGroup	Complement NB	Multinomial Decision Table NB (DTNB), Updateable NB	[22]
English	Twitter	BoW, Char n-grams	SVM, LR, CNN	[15]
English	YouTube	BoW, Word n-grams (2,3,5), Lexical Syntactic Feature	SVM, NB	[16]
English	Twitter	Word unigram	SVM, CNN, BiLSTM	[17]
English	Twitter	Word n-grams (1-8)	SVM	[18]
English	Twitter	Latent Dirichlet Allocation (LDA)	LR	[21]
English	User-generated online comments	Char and Word n-grams	NB, SVM	[14]
English	Twitter	BoW, char n-gram (3-8), word n-grams (1-3)	CNN, RNN, RF, NB, SVM, Gradient Boosted Trees, LR,	[19]
English	Twitter	BoW, word n-grams, hate or non-hate words list	SVM (linear, polynomial, radial)	[23]
English	Twitter, Articles	Abusive and non-abusive word list	Unsupervised learning	[24]
English, Portuguese	Twitter, Blogs	hateword2vec, hate-doc2vec, unigram	NB, SVM	[5]
Arabic	YouTube	Word n-grams	SVM	[6]

[11] https://tribune.com.pk/story/1300546/maryam-nawaz-forcingpml-n-leaders-abuse-public-imran

Indonesian	Twitter	cha and Word n-grams	SVM, NB, RF	[7]
Danish, English	Twitter, Facebook, Reddit	BoW, cha n-grams	LR, BiLSTM	[8]
German	Twitter	Twitter and Wikipedia embedding	CNN	[9]
Japanese	Blogs	Word n-grams (1-5)	SVM	[10]

Such events can induce a wave of anger in supporters of one political party towards other political parties. Thus, this increases the chances for malevolent tweet writing and social violence. Therefore, this period was chosen to extract maximum abusive tweets in the Urdu language.

In the crawling process, 55600 tweets in the Urdu language were retrieved that contained the seed words. The seed words are referred to as the words that were used to crawl the tweets. Although Urdu belongs to the Indo-Aryan language group, and some people believe that Urdu is a camp language, the report[12] contracted that Urdu is a camp language. Nonetheless, Urdu has roots[13] in the Arabic, Persian, and Turkish languages. Therefore, we removed all the crawled tweets that were written in other languages, such as Arabic, Persian, and Turkish. Thus, 47,700 tweets were obtained after the flirtation process, which were sent for the annotation process. In addition, instructions with a task definition and examples were provided to the annotators, particularly concerning the binary class annotation.

3.2 Dataset Normalization

In the dataset normalization process, all the non-Urdu tweets were deleted. Nonetheless, Urdu has roots[14] in the Arabic, Persian, and Turkish languages. Moreover, Urdu contains similar alphabets to these languages. Many tweets were crawled in these languages due to the same hashtags. Therefore, we removed all the crawled tweets that were written in other languages, such as Arabic, Persian, and Turkish. In addition, irrelevant tweet attributes like username, location, date and time, punctuation, uniform resource locator (URL), address, hashtag, emoticons (emojis), and the re-tweet symbol were also removed normalize the dataset and keep only the relevant information. Thus, 47700 tweets were obtained after the flirtation process, which were sent for the annotation process.

12 https://www.dawn.com/news/681263/urdusorigin-its-not-acamp-language

13 https://www.ucl.ac.uk/atlas/urdu/language.html

14 https://www.ucl.ac.uk/atlas/urdu/language.html

3.3 Guidelines for Data Annotation

This study used paid crowdsourcing to label the dataset. This study did not use Amazon Mechanical Turk for crowdsourcing; instead, the Fiverr platform was used to hire annotators, and the annotation process was completed within two months. Moreover, a digital framework was introduced to alleviate human mistakes and accelerate the dataset annotation process. A strict criterion was constructed to recruit annotators:

(a) Indigenous to Pakistan

(b) Familiar with Twitter

(c) Urdu native

(d) Dissociate from any social, profit, political, non-political party or organization

(e) The annotator should fall within the age group of 20-35 years.

These points were significantly considered to minimize annotation prejudice, especially to annotate politics or election campaign tweets. Furthermore, 16 annotators were recruited for the dataset annotation, which contained 8 males and 8 females: 10 annotators belonged to the age group of 21-25 years, while 4 annotators were between 26-30 years age group, and 2 annotators belonged to 31-35 years. We also considered the educational background of the annotators so that the bias in the dataset could be minimized. The educational background of the annotators (last degree obtained) was as follows: 8 annotators held a bachelor's degree, 4 annotators had a master's degree, and 4 annotators earned a specialized journalism degree.

Forty-seven thousand seven hundred tweets were given to the annotators, and only 3500 tweets fulfilled the annotation process. Thus, the 3500 tweets were annotated as abusive tweets. Tables 1 and 2 show a sample tweet annotated as abusive and non-abusive, respectively.

– **Abusive Tweet**: A Twitter post containing words to embarrass or humiliate other Twitter users.

– **Non-Abusive Tweet**: A Twitter post published for other objectives, such as mockery, joke, advertise, undermine, phishing, threatening, sarcasm, etc.

Word	Original Tweet
بیغیرت	بیغیرت انسان کتے کے بچے
English Translation	
Shameful	Shameful person son of dog

Table 1
Abusive tweet

Word	Original Tweet
بکواس	میری جان کیوں بکواس کرتے ہو
English Translation	
Nonsense	My love why do you talk nonsense

Table 2
Non-abusive tweet

3.4 Inter-Annotator Agreement

Calculating the agreement between dataset annotators is crucial for many reasons. First of all, this helps to annotate the dataset correctly. Secondly, bias in the dataset can be mitigated by using the inter-annotator agreement. Therefore, we used Cohan's Kappa Coefficient to quantify the reliability between annotators. As a result, a Kappa coefficient of 90% was accomplished after computing the inter-annotator agreement to collect the first abusive tweets dataset in the Urdu language.

3.5 Dataset Statistics

As a result of the annotation process, 3,500 tweets secured 100% annotator agreement for either abusive or non-abusive (and non-threatening) labels. This rigorous annotation procedure ensured the construction of a reliable dataset of 1,750 offensive tweets and 1,750 non-abusive tweets. Tables 2 show dataset statistics.

Table 2
Dataset statistics

Dataset	Words	Char	Avg Word	Total Tweets
Abusive	26,378	118,512	15	1,750
Non-Abusive	30,709	140,627	17	1,750
Totals	57,087	259,141	16	3,500

4 Experiment Settings

This section discusses the detailed experimental procedure to identify abusive tweets as a binary classification problem in which the task is to assign a label of whether a tweet is abusive or non-abusive. This study is based on neural networks (traditional machine learning), and non-neural networks (deep learning) approaches. Traditional machine learning algorithms, mainly supervised machine classifiers, were used. The machine learning classifiers used for automatic abusive language detection: Logistic Regression (LR), Multilayer Perceptron (MLP), Random Forest (FR), Support Vector Machine (SVM), and Adaboost classifier.

A python-based library, known as Scikit-Learn[15] library, was used to develop machine learning algorithms.

Two deep learning models are implemented with the Keras[16] library: 1-Dimensional Convolutional Neural Network (1D-CNN) and Long Short-Term Memory (LSTM). Eventually, for the training and the evaluation of the classifiers, 10-fold cross-validation was used. Figure 3 illustrates the overall methodology.

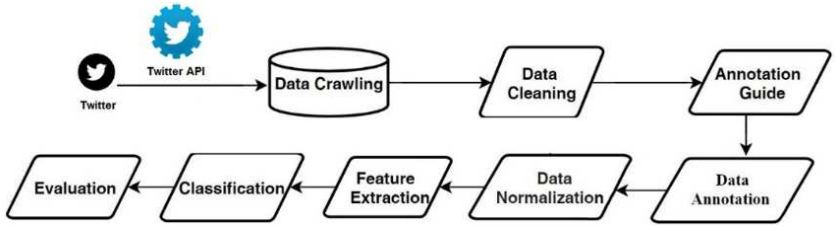

Figure 3
Methodology framework

4.1 Pre-Processing

The dataset was pre-processed to use for the experimental setup. First of all, all the tweets were converted into words (Tokens) using the white space character. Moreover, Western Arabic numerals were used to convert the numerals that followed the Eastern Arabic-Indic numeral system to normalize the entire data. Furthermore, stop words, white space tokens (blanks), punctuation, and bullets were also discarded to clean the dataset. Finally, we removed invalid utf-8 characters in the dataset and used standard utf-8 codification.

4.2 Features Extraction

This study used different text features to investigate the effect of the different types of text representation, namely, count-based (word n-grams and char n-grams) vs. embedding-based features on automatic abusive detection tasks. We considered character n-grams, word n-grams, and their combinations. We generated n-grams up to 6-grams because the previous study [4] reported that higher n-grams show insignificant results. To convert words into numeric features, a TF-IDF weighting scheme was used. We used the maximum number of features when the n-gram space dimension was higher.

For the pre-trained embedding features, we used fastText [26]. It is a neural network that is based on a word2vec algorithm. The word2vec model considers sub-words rather than dealing with entire words. In the training phase of word embeddings, if a word is not present in the dataset, its embeddings can be created by splitting the word into character n-grams.

[15] https://scikit-learn.org/stable/ [16]https://keras.io/about

4.3 Machine Learning Algorithms

For experiments, this research used five machine learning classifiers with all feature types: Multilayer Perceptron (MLP), AdaBoost, Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM). We used the default parameters of all machine learning classifiers.

4.3.1 Logistic Regression

Logistic regression (LR) is a linear model that assumes a linear relationship between input and output. It is based on the sigmoid function that measures a categorical dependent variable (abusive or non-abusive). Moreover, different studies [15, 20, 21] reported that LR provides better results on binary classification problems, particularly automatically detecting abusive language [8, 19].

4.3.2 Random Forest

A random forest (RF) is used for classification and regression tasks based on ensemble learning techniques. It uses bagging and feature randomness on different samples to construct multiple decision trees. It combines these decision trees to create a forest of trees where prediction by the majority is helpful to make an accurate prediction compared with any individual tree. The majority voting of multiple decision trees is used for the classification task, while an average of these decision trees is used in regression problems [19]. This algorithm uses dataset features to build individual decision trees and address variance and over-fitting challenges [13].

Moreover, these features are randomly selected to construct multiple decision trees. Recent studies used a random forest (RF) to classify abusive language [7].

4.3.3 Support Vector Machine

Support Vector Machine (SVM) [27] creates a line or a hyperplane (decision boundary) to separate the data into classes. It is a predictive analysis data classification algorithm used for linear, nonlinear classification, and regression tasks. Moreover, the kernel trick transforms data and finds an optimal boundary (clear margin of separation) between the possible outputs based on data transformation. SVM provides better results in high-dimensional spaces. In other words, SVM effectively performs when the dimensions are higher than the dataset instances [7]. Furthermore, different studies [5, 18, 19] reported that the SVM algorithm outperformed other classifiers in automatically identifying abusive language [10].

4.3.4 Ada-Boost

The boosting algorithm [28] is a supervised machine learning algorithm that combines different algorithms and re-assigns the weights to the input data. For the Ada-Boost algorithm, misclassified instances are crucial because the task is to make

a robust classifier by combining different algorithms to make accurate predictions. Moreover, the adaptive boosting algorithm provides higher weights only to relevant features. One of the limitations of using the AdaBoost algorithm is over-fitting. This is due to the noise present in the dataset. In other words, if features are not relevant (noisy dataset), Adaboot will not make accurate predictions. Different studies [29] revealed that using the Adaptive Boosting algorithm can effectively detect abusive language compared to other machine learning algorithms.

4.3.5 Multilayer Perceptron

A multilayer perceptron (MLP) [30] is a feed-forward neural network that generates outputs from a set of inputs. To train the MLP model, a technique called back-propagation is used that assigns weights to the neurons present in the neural network. Furthermore, this neural network consists of three layers (i) an input layer, (ii) a hidden layer, and (iii) an output layer, which is fully connected. The dataset samples are given as inputs in the input layer. Then, the dot product is used between the input samples and the weights to input the hidden layer. The output is given as input to the activation function so that the final output of the hidden layer is obtained. In the last stage, the dot product of the output of the activation function and the weights are measured, which are fed to the final layer to predict the final output. A recent study also reported that MLP showed good performance in classifying abusive language [31].

4.4 Deep Learning Classifiers

CNN and RNN are used to investigate abusive language detection tasks in Urdu. Figure 4 shows the information of deep learning parameters: all layers, parameters, and their values used for the experiments.

4.4.1 Convolutional Neural Network

A convolutional neural network (CNN) is a deep learning neural network used to solve various classification problems. This neural network typically comprises several layers where every hidden layer in the neural network contains neurons and biases. In addition to this, the dot product of the samples and the weights in the input layer is given to the activation function, which is present in the second layer. The activation function in individual neurons measures the dot product of the input samples and their weights and then adds a bias to the weighted sum. Like Multilayer Perceptron, Convolutional Neural Networks also use back-propagation to build a neural network. It is essential to mention that back-propagation reduces the error by re-assigning different values to the weights in each layer, starting from the final layer to the first layer of the neural network [17]. Moreover, this neural network is also effective in memory consumption along with dimensionality reduction.

Initially, CNN was used for image processing tasks (2D data) and video processing tasks (3D matrix). However, recent studies [9, 15, 19] also used CNN for text

classification tasks using text-based features (1D matrix) [17]. CNN is extremely efficient in extracting relevant and distinctive features and making accurate predictions. Furthermore, unlike feed-forward networks, Convolutional Neural Network is computationally efficient. Figure 5 shows the architecture of 1D-CNN that was used for automatic abusive language detection.

Parameter	1D-CNN	LSTM
Epochs	100	150
Optimizer	Adam	Adam
Loss	mean squared error	mean squared error
Learning Rate	0.0001	0.0001
Regularization	0.001	-
Bias Regularization	0.0001	-
Validation Split	0.1	0.1
Hidden Layer 1 Dimension	16	16
Hidden Layer 1 Activation	linear	tanh
Hidden Layer 1 Dropout	0.2	0.2
Hidden Layer 2 Dimension	32	16
Hidden Layer 2 Activation	linear	tanh
Hidden Layer 2 Dropout	0.2	0.2

Figure 4
Deep learning parameters

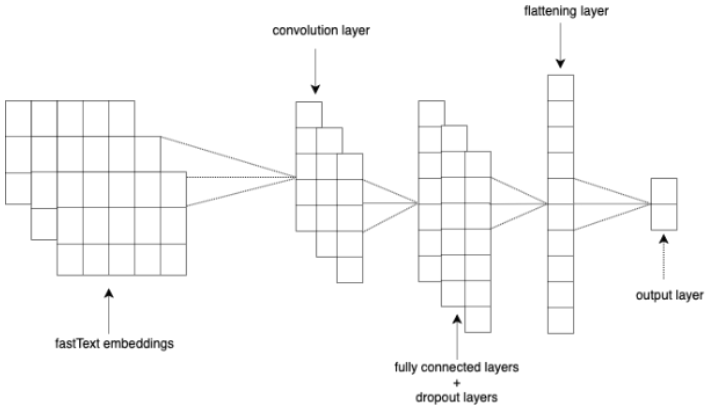

Figure 5
CNN model architecture

A pre-trained embeddings model, known as fastText embedding, is extracted from Urdu tweets to train the convolutional neural network. Subsequently, the 1D-CNN classifier receives these embeddings as input. The convolutional neural network contained two fully connected and a convolution layer. The filter size in the convolution layer was set to 8, and the window size of the kernel was fixed to 1. Moreover, this neural network is trained ten times using 100 epochs, and to avoid overfitting, a dropout is employed in all layers of the neural network. The mean accuracy of 10 iterations is used to acquire the CNN results in identifying abusive tweets.

4.4.2 Long Short-Term Memory Networks

Another deep learning algorithm, known as Long Short-Term Memory (LSTM) [8], was introduced to tackle the limitation of order dependence in sequence prediction projects, like speech recognition and machine translation [17, 19].

The LSTM model was also trained on fastText embedding extracted from Urdu tweets like the convolutional neural network. The LSTM contained two fully dense layers, and for training, each iteration had 150 epochs, and 10-fold cross-validation was employed. Initially, all the word vectors are normalized after each update, and a dropout layer between the hidden and output layer is used. The architecture of our proposed LSTM model is shown in Figure 6.

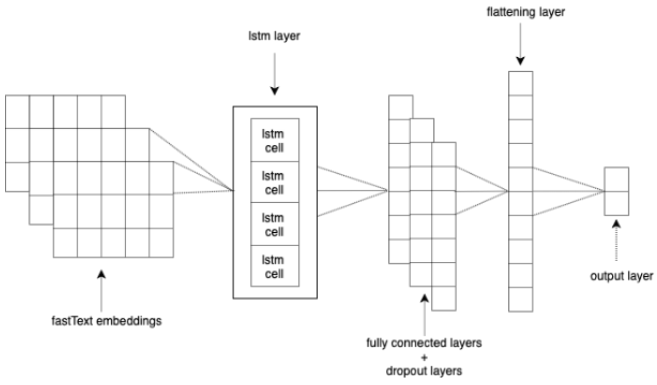

Figure 6
LSTM model architecture

4.5 Evaluation Metrics

This is a balanced dataset. For comparative analysis, two types of techniques were used in the study. For machine learning, five machine learning algorithms were used. Two neural networks were employed in the experimental setting for deep learning. Therefore, the algorithm's selection is deemed to be appropriate for this task. All the proposed models are evaluated based on standard metrics, including F-measure, accuracy, and (ROC) curve.

5 Results and Analysis

We ran experiments using three text representations: two count-based features (word and char n-grams) and pre-trained fastText word embeddings. This study used fastText embeddings because this embedding represents each word as the sum of the n-gram vectors rather than learning vectors for each word. This embedding contains word vectors for 157 languages learned on Wikipedia and Crawl and addresses the issue of out of vocabulary (OOV). For example, boxer and boxing are employed in distinct contexts, and capturing the underlying commonality of both words is challenging. Therefore, this embedding addresses this problem by dividing the words into character n-grams. Furthermore, as compared to the BERT, fastText embeddings are exceptionally quick and can be trained on more than one billion words in less than ten minutes using a normal multicore CPU.

Moreover, we generated n-grams up to 6-grams for words characters and words. However, the results of word n-grams started to decrease after 4-grams. This is why the results of 5 and 6-word grams are not provided. The results of character n-grams, fastText, and word n-gram are shown in Tables 4, 5, and 6, respectively. The feature column in these tables represents the maximum number of features used to train the machine learning classifiers to distinguish abusive and non-abusive tweets. Moreover, character n-grams and word n-grams were extracted using the TF-IDF weighting scheme.

The results show that SVM outperformed other classifiers and achieved the highest accuracy of 82.37% and F_1 score of 82.68% with char tri-gram features. We only investigated the linear kernel and noticed that its performance was sufficiently high for the baseline experiments. Moreover, LR performed best on all char n-gram features. On the other hand, RF performed worst on the same features. Furthermore, SVM also achieved the best results using word unigram features, slightly less than the highest results. It had an F1 score of 81.85% and an accuracy of 81.27%. However, all the other machine learning models performed worst on bi-gram, tri-gram, and the combination of word n-gram features. Figure 7 illustrates the ROC curve, and Figure 8 represents the confusion matrix of SVM to differentiate between

abusive and non-abusive tweets.

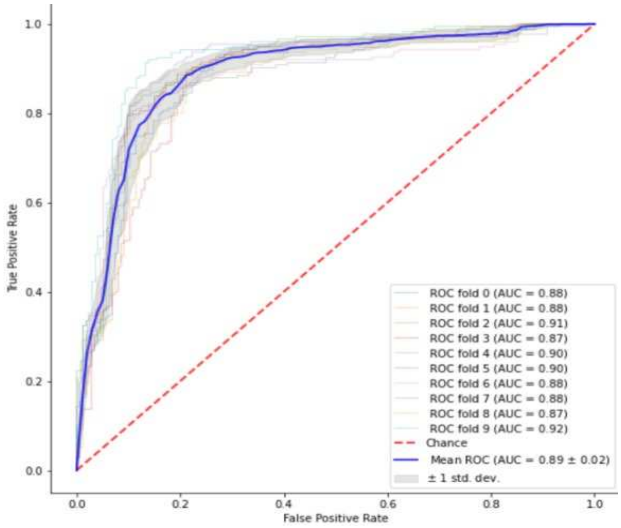

Figure 7
ROC curve for best performing model (SVM)

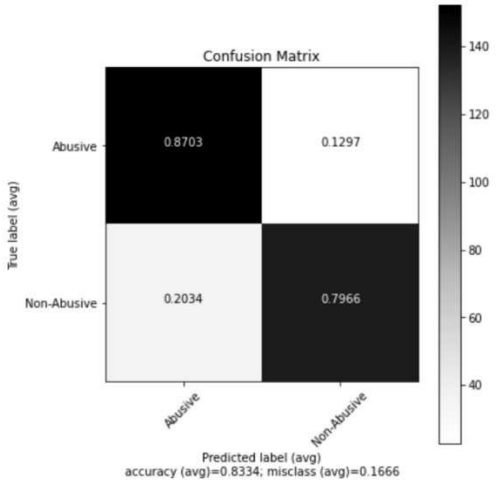

Figure 8
Confusion matrix for best performing model (SVM)

Deep learning algorithms like 1D-CNN and LSTM using fastText pre-trained word embeddings could not achieve the highest results for abusive language detection. Because of the limited training data, most of the words are not present in the fastText vocabulary. Another primary reason is that we used random vectors for out-of-vocabulary words. As a result, most of the vectors were diluted with bias. Moreover,

it seems that the performance of deep learning classifiers improves as the dataset size increases. Overall, our results align with state-of-the-art efforts in abusive language detection and illustrate that there is still a great deal of room for growth.

Conclusions

Automatic threat language detection in English or other European languages is a challenging task that is widely examined. Nonetheless, as far as we can ascertain, there is no investigation into automatic abusive language detection in Urdu using Twitter postings. This paper contains a two-fold contribution. First, we collected and annotated the first corpus of tweets in the Urdu language for automatic abusive language detection. The corpus contained 3500 tweets that passed through pre-processing and rigorous manual annotation. Second, we compared the potential of various text representations for automated abusive language detection in Urdu tweets and ran a series of experiments with five different classification algorithms.

The experiment results demonstrate that SVM consistently obtained improved outcomes for both count-based feature types; the word unigrams and character tri-gram got better results than other n-gram features. Moreover, the fastText pre-trained word embeddings for Urdu obtained comparatively low results than the n-gram features. It might be because of the limited corpus size required to pre-train the embedding model, as well as a high number of out-of-vocabulary words that are likely to be present in abusive tweets. These baseline results will serve as a reference point for evaluating classification techniques developed by other researchers in the future. We aim to increase the dataset size and use transformers-based techniques to address abusive language detection in Urdu using Twitter postings for future research.

Table 4
Identification of abusive tweets using char-level features (TFIDF-based)

Feature set	# of Features	–	Classifiers				
			LR	MLP	AdaBoost	RF	SVM
3-gram	9491	P	86.51	79.00	83.31	83.39	86.09
		R	78.68	77.48	74.85	81.14	79.65
		Acc	83.17	78.40	79.85	82.45	83.34
		F_1	82.37	78.14	78.73	82.18	82.68
4-gram	29,138	P	87.03	80.44	85.67	83.17	86.16
		R	78.45	77.82	76.80	81.25	77.88
		Acc	83.37	79.40	81.94	82.37	82.65
		F_1	82.47	79.04	80.94	82.14	81.75
5-gram	59,460	P	86.66	80.26	86.30	81.25	86.23
		R	77.42	79.65	75.77	82.28	75.60
		Acc	82.74	80.00	81.85	81.62	81.74
		F_1	81.75	79.91	80.67	81.73	80.51
6-gram	90,573	P	86.62	76.67	86.06	77.11	86.45

		R	74.00	81.31	70.17	81.42	71.02
		Acc	81.25	78.22	79.37	78.57	79.94
		F_1	79.77	78.87	77.24	79.16	77.95
combination (3-6)-gram	188,662	P	86.82	81.88	85.23	84.17	86.17
		R	78.00	78.22	75.82	80.80	76.97
		Acc	83.05	80.42	81.28	82.77	82.28
		F_1	82.14	79.95	80.14	82.41	81.27

Table 5

Identification of abusive tweets using word-level features (TFIDF-based)

Feature set	# of Features	–	Classifiers	
			1D-CNN	LSTM
fastText	300	P	79.47	79.45
		R	79.37	77.42
		Acc	79.42	78.68
		F1	79.39	78.39

Table 6

Identification of abusive tweets using word-level features (TFIDF-based)

Feature set	# of Features	–			Classifiers		
			LR	MLP	AdaBoost	RF	SVM
unigram	6,671	Acc	82.31	77.40	81.25	81.25	82.82
		P	86.30	77.60	87.49	84.13	86.65
		R	76.85	77.08	72.97	77.08	77.65
		F_1	81.27	77.26	79.54	80.40	81.85
bigram	28,929	Acc	76.42	74.71	68.02	70.42	74.17
		P	83.65	72.23	89.23	66.32	85.19
		R	65.77	80.51	41.08	83.20	58.51
		F_1	73.60	76.09	56.14	73.76	69.32
trigram	38,006	Acc	69.11	58.05	54.57	56.40	65.45
		P	81.60	55.03	83.68	53.82	81.63
		R	49.37	88.40	11.31	90.11	39.94
		F_1	61.43	67.82	19.79	67.39	53.54
4-gram	37,577	Acc	64.54	51.42	52.94	51.48	64.57
		P	80.60	50.78	51.52	50.81	79.71
		R	38.34	90.39	99.42	92.57	39.20
		F_1	51.92	65.03	67.87	65.60	52.46
combination (1-4)-gram	111,183	Acc	81.25	80.05	81.34	79.40	78.02
		P	86.55	82.16	87.33	85.20	86.77
		R	74.05	76.91	73.31	71.25	66.17
		F_1	79.77	79.38	79.68	77.55	75.03

Acknowledgments

The work was supported by the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico and grants 20211784, 20211884, and 20211178 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors utilize the computing resources provided by the CONACYT through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico.

References

[1] W. Schwartz: Descriptive Psychology, and the Person Concept*:* Essential Attributes of Persons and Behavior, Academic Press, 2019

[2] Y. Urano, R. Takizawa, M. Ohka, H. Yamasaki, H. Shimoyama: Cyberbullying victimization and adolescent mental health: The differential moderating effects of intrapersonal and interpersonal emotional competence, Journal of Adolescence, Vol. 80, 2020, pp. 182-191

[3] Y. Zhu, W. Li, J. E. O'Brien, T. Liu: Parent-child attachment moderates the associations between cyberbullying victimization and adolescents' health/mental health problems: An exploration of cyberbullying victimization among Chinese adolescents, Journal of Interpersonal Violence, 2019

[4] M. Amjad, G. Sidorov, A. Zhila, H. Gómez-Adorno, I. Voronkov, A. Gelbukh: Bend the truth: Benchmark dataset for fake news detection in Urdu language and its evaluation, Journal of Intelligent & Fuzzy Systems, Vol. 39, No. 2, 2020, pp. 2457-2469

[5] R. Pelle, C. Alcântara, V. P. Moreira: A classifier ensemble for offensive text detection, Proceedings of the 24th Brazilian Symposium on Multimedia and the Web, 2018, pp. 237-243

[6] A. Alakrot, L. Murray, N. S. Nikolov: Towards accurate detection of offensive language in online communication in Arabic, Procedia Computer Science, Vol. 142, 2018, pp. 315-320

[7] M. O. Ibrohim, I. Budi: A dataset and preliminaries study for abusive language detection in Indonesian social media, Procedia Computer Science, Vol. 135, 2018, pp. 222-229

[8] G. I. Sigurbergsson, L. Derczynski: Offensive language and hate speech detection for Danish, arXiv preprint arXiv:1908.04531, 2019

[9] J. M. Schneider, R. Roller, P. Bourgonje, S. Hegele, G. Rehm: Towards the automatic classification of offensive language and related phenomena in German tweets, 14th Conference on Natural Language Processing KONVENS, 2018, p. 95

[10] T. Ishisaka, K. Yamamoto: Detecting nasty comments from BBS posts, Proceedings of the 24th Pacific Asia Conference on Language, Information

and Computation, 2010, pp. 645-652

[11] G. Sterner, D. Felmlee: The social networks of cyberbullying on Twitter, International Journal of Technoethics (IJT), Vol. 8, No. 2, 2017, pp. 1-15

[12] D. Chatzakou, N. Kourtellis, J. Blackburn, E. D. Cristofaro, G. Stringhini, A. Vakali: Mean birds: Detecting aggression and bullying on Twitter, Proceedings of the 2017 ACM on Web Science Conference, 2017, pp. 13-22

[13] V. Balakrishnan, S. Khan, T. Fernandez, H. R. Arabnia: Cyberbullying detection on Twitter using Big Five and Dark Triad features, Personality and Individual Differences, Vol. 141, 2019, pp. 252-257

[14] Y. Mehdad, J. Tetreault: Do characters abuse more than words?, Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2016, pp. 299-303

[15] J. H. Park, P. Fung: One-step and two-step classification for abusive language detection on Twitter, arXiv preprint arXiv:1706.01206, 2017

[16] Y. Chen, Y. Zhou, S. Zhu, H. Xu: Detecting offensive language in social media to protect adolescent online safety, 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing, 2012, pp. 71-80

[17] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar: Predicting the type and target of offensive posts in social media, arXiv preprint arXiv:1902.09666, 2019

[18] P. Rani, A. K. Ojha: KMI-coling at SemEval-2019 task 6: exploring N-grams for offensive language detection, Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 668-671

[19] Y. Lee, S. Yoon, K. Jung: Comparative studies of detecting abusive language on Twitter, arXiv preprint arXiv:1808.10245, 2018

[20] P. Mishra, M. D. Tredici, H. Yannakoudakis, E. Shutova: Abusive language detection with graph convolutional networks, arXiv preprint arXiv:1904.04073, 2019

[21] G. Xiang, B. Fan, L. Wang, J. Hong, C. Rose: Detecting offensive tweets via topical feature discovery over a large-scale twitter corpus, Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012, pp. 1980-1984

[22] A. H. Razavi, D. Inkpen, S. Uritsky, S. Matwin: Offensive language detection using multi-level classification, Canadian Conference on Artificial Intelligence, Springer, Berlin, Heidelberg, 2010, pp. 16-27

[23] P. Burnap, M. L. Williams: Us and them: identifying cyber hate on Twitter across multiple protected characteristics, EPJ Data Science, Vol. 5, 2016, pp. 1-15

[24] H. S. Lee, H. R. Lee, J. U. Park, Y. S. Han: An abusive text detection system based on enhanced abusive and non-abusive word lists, Decision Support Systems, Vol. 113, 2018, pp. 22-31

[25] J. Cohen: A coefficient of agreement for nominal scales, Educational and Psychological Measurement, Vol. 20, No. 1, 1960, pp. 37-46

[26] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov: Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics, Vol. 5, 2017, pp. 135-146

[27] N. Rusnachenko, N. Loukachevitch, E. Tutubalina: Distant supervision for sentiment attitude extraction, Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), 2019, pp. 1022-1030

[28] C. Ying, M. Qi-Guang, L. Jia-Chen, G. Lin: Advance and prospects of adaboost algorithm. Acta Automatica Sinica, Vol. 39, No. 6, 2013, pp. 745-758

[29] R. E. Schapire: Explaining Adaboost, Empirical Inference, Springer, Berlin, Heidelberg, 2013, pp. 37-52

[30] D. W. Ruck, S. K. Rogers, M. Kabrisky: Feature selection using a multilayer perceptron, Journal of Neural Network Computing, Vol. 2, No. 2, 1990, pp. 40-48

[31] M. W. Gardner, S. R. Dorling: Artificial neural networks (the multilayer perceptron): A review of applications in the atmospheric sciences, Atmospheric Environment, Vol. 32, No. 14-15, 1998, pp. 2627-2636

[32] Amjad. M, Zhila. A, Sidorov. G, Labunets. A, Butt. S, Amjad. H. I, Vitman. O, Gelbukh. A: UrduThreat@ FIRE2021: Shared Track on Abusive Threat Identification in Urdu, In Forum for Information Retrieval Evaluation, 2021, pp. 9-11

Automatic Detection of Opposition Relations in Legal Texts Using Sentiment Analysis Techniques: A Case Study

Obdulia Pichardo-Lagunas[1], Bella Martinez-Seis[1], Miguel Hidalgo-Reyes[2], Sabino Miranda[3]

[1] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, Avenida Instituto Politécnico Nacional 2580, Barrio la Laguna Ticomán, Gustavo A. Madero, 07340, Ciudad de México, México;opichardola@ipn.mx, bcmartinez@ipn.mx

[2] Instituto de Ecología A.C. (INECOL), Carretera antigua a Coatepec 351, Col. El Haya, Xalapa, Veracruz. CP 91073, México; miguel.hidalgo@inecol.mx

[3]INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Circuito Tecnopolo Sur 112, Fracc. Tecnopolo Pocitos, 20313, Aguascalientes, México; sabino.miranda@infotec.mx

Abstract: The documentation that describes the regulations within a Society, is oriented towards specific areas. This fact does not prevent maintaining concordance in the temporality and transversality of the documents. This work defines the concept of "opposition relations" in legal texts. We identify entities and evaluate the polarity of each paragraph with sentiment analysis techniques. If an entity appears in different paragraphs (articles of law) with opposite polarities, we evaluate the entity's contexts. We look for antonyms between the words that give polarity to the opposite paragraphs. If there is an antonymic relation in words associated with the entity, we have an opposition relation. The described methodology analyzes the relationship of entities in Mexican Environmental Laws, and the study is oriented towards coherence in the legislation for sustainable development. This process was implemented by computational processing, which required the transformation of current Mexican laws, unifying its structure. Eight environmental laws were analyzed, 1920 entities were identified that appear more than once; 44 of them were identified with opposite polarities, due to their context, a detailed analysis of two cases with potential opposite relationships is exemplified.

Keywords: Natural Language Processing; Legal Text; Sentiment Analysis; Opposition Relation

1 Introduction

The generation of laws and regulations at different countries, states, and municipalities allows implementing public policies to benefit the societies they relate. Laws and regulations are written with specific objectives aimed at specific scopes and concerning the different actors involved. For instance, in Mexico, the *Ley de Fondos de Inversión* (Investment Funds Law) is mainly oriented toward the economic sector. In contrast, the *Ley de Hidrocarburos* (Hydrocarbons Law) draws together actors from economics, politics, international, environmental, and transport sectors, among others.

In the last decade, creating public policy legislation that considers the concept of sustainable development has been one of the objectives set by different governments, together with agreements reached with organizations such as the OECD (Organization for Economic Cooperation and Development).

By legislating in this sense, it is intended to put the environment as a priority issue and its protection as a necessary condition for the construction of the development of any country [1]. By directing the actions of federal public administration agencies and entities towards the concept of sustainable development, it is intended to permeate this term throughout the legal system [2] achieving an instrument of truly transversal nature that establishes principles of observable and punishable performance in all activities [3].

However, in many cases, the writing on the legislative documents do not maintain the expected coherence between proposed public policies and the sustainable development goals promised by the State. From this environmental legislation perspective arises the motivation of this work, we make a computational analysis of the laws in order to measure the coherence level between the design of Mexican public policies embodied in environmental legislation and the concept of sustainable development.

The present work proposes a methodology that, using techniques of Natural Language Processing (NLP), identifies the possible inconsistencies in a corpus of laws and regulations from the Mexican normative, based on the different entities identified. This methodology identifies entities and analyzes the paragraphs that contains them. A sentiment analysis is applied to identify the polarity (positive or negative) in the named entities, highlighting opposition relations, also considering the antonymic relations.

We define "opposition relation" as the association that exists between two text sections, from the same or different documents, that are evaluated with opposite polarities while sharing actors and that, in the context of identified entities, contain vocabulary with antonymic relations.

The hypothesis of this work is as follows: The opposition relations identify potentially incoherent elements, because based on the polarity assigned to the context of a previously identified actor (entity) and shared across different sections

of text, it is expected that the classification stays the same. That is, if two texts share an entity, both should be classified with the same polarity and the vocabulary that contextualizes them should not keep antonymic relations.

By developing this type of methodologies, computational tools that help in the validation of the coherence relation between the writing of public policies and sustainable development goals could be generated.

It is difficult to carry out an automatic computational analysis in texts pertaining the legal field, as it is important, initially, to identify characteristics that allow to build computational structures that enable a syntactic analysis. However, given the complexity of the task and the nature of the texts, there are no tools at the moment that allow this task to be carried out properly. Some of the problems that can be identified are:

1. To identify patterns at legal documents that allow to discover knowledge.
2. To identify the semantics of some of the words used in the legal scene, since it sometimes differs from their semantics in a general scope, i.e., how they are commonly used.
3. To build knowledge bases that allow to represent the different concepts within the legal scene.

2 NLP: Overview of Legal Texts

Various applications have been made in the legal framework by using Natural Language Processing (NLP) and Artificial Intelligence (AI) [4]. Standing out among them are the prediction of trials [5], question answering [6], searches [7] and text summarization in various legal frameworks [8]. However, some analyzes represent interesting opportunities to explore, for example, measuring the level of coherence between laws.

According to Zhong et al. [9], two types of law processing methods stand out: on the one hand, methods based on symbols that use an interpreted knowledge or the learned rules [10]; on the other hand, embedded methods that learn on a large scale. The former has a lower efficiency while the latter lacks interpretability, which represented an obstacle in this work due to the validation condition.

The processing of laws with artificial intelligence techniques has been used in multiple works analyzing different countries and languages. For example, in [5] is analyzed the results of various labor lawsuits in different Brazilian regions using different machine learning techniques, identifying the reasons why a trial turns in favor or against. Meanwhile, in [11], the authors classify judgments based on German legislation by using machine learning, classifiers like Support Vector Machines (SVM) or Naive-Bayes, identifying classes as definitions, obligations,

revocations (repeal), among others. They build a pattern-based classifier by the comparison of candidate phrases with standard phrases, with around 90% of accuracy; they point out that when writing German laws, they use legal conventions that allow the interpretation of similar information, using structures such as by *X* is understood *Y* where *X* is a term and *Y* a definition. In addition, Son et al. [12] present an analysis for laws in the Vietnamese language using deep neural networks.

2.1 Structure of Legal Documents

Among the problems identified by several authors, there is a lack of standardization when writing a law.

Sometimes, the expected pattern is not fitted, or the expressiveness of the words found is inadequate since they could belong to different classes. Palmirani's proposal [13] presents a structure in XML to describe legislative documents under two standards. The general structure includes elements of the normative act, particular elements such as references, and meta-information like references to other laws, notes, or locations.

Ashley and Rissland [14] focus on constructing a computational representation of laws based on legislations from the US and Great Britain. They identify rules and actions from different legislations and the word sets or corpus over time.

The task of recognizing the logical compositions in legal documents ranges from assumptions to sanctions, elements that help to detect a legal document structure. In Mexico, there is no standardization in the structure of laws, either by temporality or by changes in the executive branch, which complicates generating a computational structure.

2.2 Identification of the Coherence Level

From the perspective of identifying the level of coherence in legal documents, in [15], the authors present a project developed by Stanford University in which, by using a software tool, they structure the knowledge that is found within a set of legal texts of the United States. Based on different representations of knowledge extracted using the mentioned tool and labeled manually, among other tasks, they focus on identifying inconsistencies, highlighting the importance of the generation of interpretive manuals due to the difficulty of analyzing legal texts given the interpretation to which they are subject.

In [16] the authors identify the orders that are most closely related by a domain search and organizational structures, giving a measure of similarity. This similarity can be used to measure the coherence between documents.

3 Text Analysis Techniques

The use of computer tools in analyzing texts from any domain is already known; however, the use of the internet has allowed access to large collections of legal texts from various topics and countries. It has increased the scientific community's interest in processing, analyzing, and recovering text information from legal documents. One of the promising research areas is based on acquiring legal knowledge and the synthesis techniques of documents and hypertext structures.

3.1 Named Entity Recognition

Named Entity Recognition consists in locating and classifying lexical units by studying categories such as places, people, organizations, time, and quantities expressions. It is recognized as one of the main tasks of natural language processing (NLP). For example, in the text shown in 1 the identified entities are marked with bold text. It is notorious that entities comprise one or more words that represent the element on which the action takes place.

Table 1
Named entities example in Spanish and its translation in English

ARTÍCULO 6o.- Las atribuciones que esta Ley otorga a la Federación, serán ejercidas por el Poder Ejecutivo Federal a través de la Secretaría y, en su caso, podrán colaborar con ésta las Secretarías de Defensa Nacional y de Marina cuando por la naturaleza y gravedad del problema así lo determine, salvo las que directamente corresponden al Presidente de la República por disposición expresa de la Ley.
ARTICLE 6. The attributions that this Law grants to the Federation, will be exercised by the Federal Executive Power through the Secretariat and, if necessary the Secretariats of National Defense and Navy may collaborate with it when determined by the nature and seriousness of the problem, except those that directly correspond to the President of the Republic by express provision of the Law.

Currently, there are different software tools that automatically allow the identification of entities [17]. Specifically, universities such as Stanford or the Massachusetts Institute of Technology (MIT) have developed tools to identify different types of entities; one of them is the spaCy library [18].

SpaCy is a library developed by MIT that contains pre-trained statistical models and word vectors, allowing the tokenization process in 49 languages. It has convolutional neural networks models for labeling, analysis, and recognition of named entities, integrating deep learning modules to solve these tasks.

The library allows to identify different types of entities such as PERSON (people, even fictitious ones), NORP (nationalities or religious or political groups), FAC Buildings, airports, roads, bridges, etc.), ORG (companies, agencies, institutions,

etc.), GPE (countries, cities, states) and LOC (non GPE Locations like mountain ranges, bodies of water). In the example of the 6th Article of Table II, the entities classified as ORGanization or PERSON are shown.

Table 2
Labeled entities example in Spanish and its translation in English

ARTÍCULO 6o.- Las atribuciones que esta Ley otorga a la Federación(ORG), serán ejercidas por el Poder Ejecutivo Federal(ORG) a través de la Secretaría(ORG) y, en su caso, podrán colaborar con ésta las Secretarías de Defensa Nacional y de Marina(ORG) cuando por la naturaleza y gravedad del problema así lo determine, salvo las que directamente corresponden al Presidente de la República(PERSON) por disposición expresa de la Ley.
ARTICLE 6. The attributions that this Law grants to the Federation(*ORG*), will be exercised by the Federal Executive Power(*ORG*) through the Secretariat(*ORG*) and, if necessary the Secretariats of National Defense and Navy(*ORG*) may collaborate with it when determined by the nature and seriousness of the problem, except those that directly correspond to the President of the Republic(*PERSON*) by express provision of the Law.

3.2 Sentiment Analysis (SA)

The sentiment analysis (SA) is an NLP technique also known as Opinion Mining. The SA aims to perform automatic text classification based on positive or negative connotations of the language used.

Among the tools used to solve this task include the lexicons or dictionaries [19]. The work of Zafra et al. [20] contains a list of words classified in two categories (positive or negative) that will help as a guide for the evaluation of polarity in a text. A software tool used in sentiment analysis is SentiStrength [21] [22].

SentiStrength was initially designed to assess the sentiment in published texts on MySpace [21]. However, it has been used with good results in short text analysis. This tool proposes the use of dictionaries incorporating sets of grammar rules, which provides a double orientation in the identification of the sentiment in the text. SentiStrength provides two values for the parsed text: the first one measures the intensity of the positive sentiment, and the second one the intensity of the negative sentiment. These values range from 1 to 5, also providing the result of the analysis in three formats: Binary (positive/negative), Trinary (positive/negative/neutral) and Simple scale (from -4 to +4) [23].

Negation Treatment in SentiStrength

The negation markers such as "no" (not), "ni" (neither), among others are used employing a list of words by the SentiStrength algorithm, which assigns a polarity to the paragraphs. The negating word list is used to invert following emotion words, skipping any intervening booster words; for more details, see [21] [22].

4 Methodology: Detection of Semantic Opposites in Mexican Environmental Legislation

The methodology described in this document aims to identify the opposition relations existing in a selected set of texts, based on the evaluation of polarity and antonymic relations between them. The stages that make up the solution of this task are shown in the Figure 1.

Figure 1
Described methodology for the identification of opposition

4.1 Text Selection and Entity Identification

The first stage of the methodology for the detection of opposition relations consists of the four tasks shown in Figure 2, going from the selection of laws to the identification of entities, going through the transformation of documents to semi-structured texts and formats.

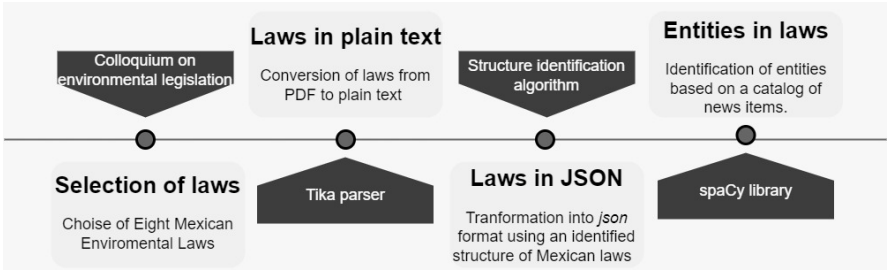

Figure 2
Text selection and entity identification (first step of methodology)

The selection of the set of legislative documents, specifically laws, consists of a set of 8 laws of environmental matter identified as the most significant at the First Interdisciplinary Colloquium for the Analysis of Environmental Legislation [24].

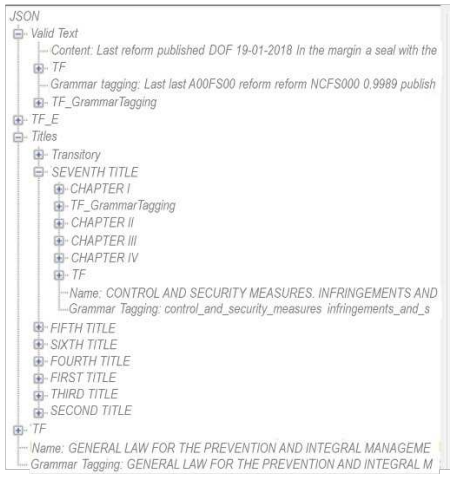

Figure 3
Example of the hierarchical structure identified in a legal document

At this stage, the text was pre-processed and stored in JSON format. The proposed format for document storage allows to maintain the natural hierarchical structure used in the drafting of regulations. Conversion from the source format of the used files was done using the Tika Python library [25].

The Tika library [25] reads the file in pdf format and extracts the metadata of the document and the corresponding content values to later write out all the information in a plain text file.

Next, an algorithm that allows the identification of a generalized hierarchical structure in Mexican laws was designed; this algorithm transforms the plain text to a semi-structured text in JSON format, customized for legislation. Figure 3 shows an example of the hierarchical structure identified in a legal document.

The entity recognition in the texts was carried out using the spaCy library [26]. This library has a catalog of entities based on identified entities in news notes. It is important to mention that the catalog used by spaCy is of common knowledge, meaning that it does not count with vocabulary specialized in legislative matter.

4.2 Sentiment Analysis of the Entity's Contexts

This stage aims to identify the polarity of the words around the detected entities as shown in Figure 4. Specifically, the polarity of the paragraphs in which the entities occurred is obtained to detect pairs of paragraphs that could have any opposition relation concerning the entities. Once the entities described in Section 4.1 have been identified, these are replaced in the generated text file assigning a sequential number to each of the entities. Each entity is replaced by the word entity plus the corresponding sequential number; for example, the entity *"Aguas Nacionales"*

(National Waters) is replaced in the documents by the term "entity45". This substitution is made to prevent in this stage that the tool used recognizes any word that belongs to the entity as a term that could indicate polarity (positive or negative).

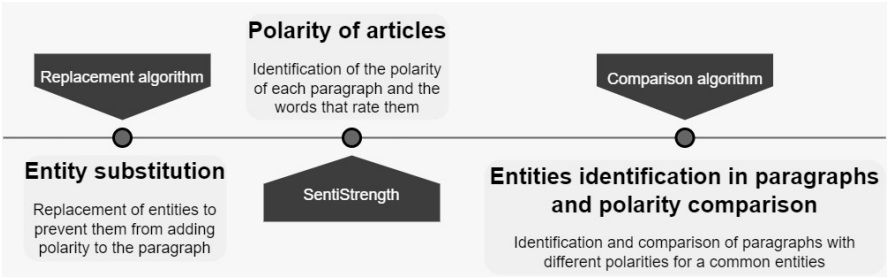

Figure 4
Process for the detection of the sentiments associated with an entity, according to the paragraphs in which the latter is located (second methodology stage)

Once the entities are substituted, each one of the paragraphs is sent to the SentiStrength tool [21], to identify each paragraph's polarity and word's polarity that indicate any charge, both positive and negative, considering the entities identified in Section 4.1 and the polarities of the paragraphs in which they are found.

Next, for a particular entity, a Python script is used to compare the polarities identifying those paragraphs that show a distinct polarity between them. That is, positive and negative polarities, an example is shown in Section 5.3.

4.3 Antonym Identification

At this stage, given the entities that show different polarities in different paragraphs in the same law or two different laws, each word from a particular paragraph that indicates any sentiment, positive or negative, is compared with the words from another paragraph that indicates the opposite sentiment, negative or positive, to identify antonyms. The identification is performed using a dictionary of antonyms based on the Wordreference platform [27], an example of this case is shown in Table 8, in which the word *"explotación"* (exploitation) is found in the first column and the second column its antonyms identified in another paragraph.

Finally, by identifying different words that indicate a polarity of different paragraphs by a relation of antonyms, these paragraphs are identified as having an opposition relationship for a particular entity.

4.4 Coherence Validation

In this stage, a list of the entities that appear in at least two separate paragraphs is obtained. The list contains the number of times that each entity appears for each polarity, that is, the number of positive, negative, and neutral paragraphs in which

each entity appears. For the entities that appear in paragraphs with different polarities, an analysis of antonyms is carried out within the words that give the meaning charge to polarity analysis. It allows to identify potential opposite relations that would generate low levels of coherence between laws.

The coherence validation is carried out by the hand of public policy experts, lawyers, and legislators. In this way, they would be able to delve into the context of entities and determine the level of coherence, using this tool as support. In other contexts, there are labeled corpora related to the terms polarity, but in the legislative and environmental context there are no developed semantic resources for legal domain.

5 Analysis of Results in Mexican Environmental Laws

This section shows some of the results obtained by applying the presented methodology on the test corpus, described in Section 4, recalling that an opposition relation is detected when antonyms between two different paragraphs of two different laws are detected.

5.1 Laws Documents

This work is developed in the environmental context, and in this particular case, eight environmental laws listed in Table 3 were analyzed.

Table 3
Document's Names employed in the experiment in Spanish and its translation in English

Mexican Environmental Laws
Ley General del Equilibrio Ecológico y la Protección al Ambiente (General Law of Ecological Balance and Environmental Protection)
Ley de Aguas Nacionales, reformada en el 2020 (National Waters Law, reformed in 2020)
Ley de Bioseguridad de Organismos Genéticamente Modificados (Biosafety Law for Genetically Modified Organisms)
Ley General de Desarrollo Forestal Sustentable (General Law of Sustainable Forest Development)
Ley General de Pesca y Acuacultura Sustentable (General Law on Sustainable Fisheries and Aquaculture)
Ley General de Vida Silvestre (General Wildlife Law)
Ley General para la Prevención y Gestión Integral de los Residuos (General Law for the Prevention and Comprehensive Management of Waste)

5.2 Identification of Potential Semantic Opposites in Environmental Laws

The methodology described in Section 4.1 obtains the list of all entities with more than one appearance to compare them. This list identifies how many times an entity appears in paragraphs with positive, negative, and neutral polarity; and which laws they appeared. An example of this list is shown in Table 4.

Table 4

Example of some detected entities in Spanish and its translation in English. It is shown the number of occurrences and the polarity associated with the paragraphs in which they appear

Entity	**Repetitions**	**Constant Polarity**	**Positive Appearances**	**Negative Appearances**	**Neutral Appearances**
Nueva ley (new law)	2	Neutral	0	0	4
Diario Oficial de la Federación (Official Journal of the Federation*)*	8	-	14	1	1
Ley de Comercio Exterior(Foreign Trade Law)	3	-	2	4	0
Estados Unidos Mexicanos (Mexico)	8	Neutral	0	0	16
Ley (Law)	7	Positive	14	0	0
Vicente Fox Quesada (name of former president)	3	Neutral	0	0	6
Presidente de los Estados Unidos Mexicanos (President of Mexico)	5	-	2	0	8
Aguas Nacionales (National Waters)	2	-	3	1	0
El servicio de aseguramiento (The assurance service)	1	Negative	0	2	0
Foro Consultivo Científico y Tecnológico (Scientific and Technological Consultative Forum)	1	-	1	1	0
Comisión Federal de Electricidad (Federal electricity commission)	2	-	3	1	0

Ley General de Pesca y Acuacultura Sustentable (General Law on Sustainable Fisheries and Aquaculture)	4	-	3	1	4
Secretarías (Secretaries)	3	-	4	0	2

In the study of the eight environmental laws, 1920 entities were identified with at least two appearances. The distribution of these entities is shown in Fig. 5. 1240 entities appear in paragraphs with constant polarity. It means that all its occurrences are positive, negative, or neutral; for example, the entity "*Estados Unidos Mexicanos*" (Mexico, full name of Mexico Country) appears 16 times in the eight laws, and all their appearances are in paragraphs with neutral polarity.

Out of the 1920 entities, only 680 do not have a constant polarity. In this sense, two levels of coherence are identified. The first refers to entities in paragraphs with neutral occurrences combined with positive or negative occurrences; in such a manner that the level of coherence could still be present; this is the case of 636 entities out of the 680. However, for the second level identified, there are entities that have occurrences in both positive and negative contexts, which could present potential opposites relations, such is the case of the 44 entities shown in red in Fig. 5. This low percentage is consistent with the fact that all the laws analyzed in this work belong to the same topic.

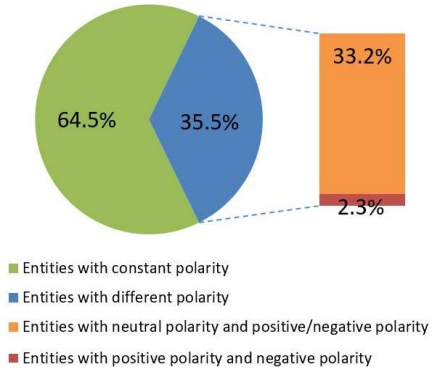

Figure 5
Polarity distribution of the 1920 entities with at least two occurrences, in the eight environmental laws

The entities that are in paragraphs with positive and negative polarity are studied in the antonym identification stage (Section 4.3) and in the coherence validation stage (Section 4.4). From those last stages, most of the entities, like "Economía" (Economy), are in paragraphs with different polarities, but there are no antonyms detected in the words that give polarity to the paragraphs; then, an opposition relation is not detected. Nevertheless, few entities present potential opposition relations, the entities that stand out are exemplified below within three of the laws.

5.3 Opposition Markings Identified for the Entity "*Aguas Nacionales*" (*National Waters)*

After analyzing the law: *"Ley General de Equilibrio Ecológico y Protección al Ambiente", LGEEPA* (General Law of Ecological Balance and Environmental Protection) and the law: "*Ley de Aguas Nacionales*", *LAN* (National Waters Law), it was identified that they share the use of some entities such as "Aguas Nacionales" (National Waters) whose paragraphs of occurrence are shown in Table 5 and 6 respectively. Based on the entity "Aguas Nacionales" (National Waters) and by analyzing the vocabulary found in the context of this entity, the existence of an opposition relation between both texts was detected.

Table 5

Paragraph of LGEEPA with positive charge for "aguas nacionales" (national waters) in Spanish and its translation in English

Los lineamientos para la realización de las acciones de preservación, restauración y aprovechamiento sustentable de los recursos naturales dentro de las áreas naturales protegidas, para su administración y vigilancia, así como para la elaboración de las reglas administrativas a que se sujetarán las actividades dentro del área respectiva, conforme a lo dispuesto en ésta y otras leyes aplicables; Las medidas que el entity41 podrá imponer para la preservación y protección de las áreas naturales protegidas, serán únicamente las que se establecen, según las materias respectivas, en la presente Ley, las entity116, de aguas nacionales, de entity149, entity58, y las demás que resulten aplicables
The guidelines for carrying out actions for the preservation, restoration and sustainable use of natural resources within protected natural areas, for their administration and surveillance, as well as for the elaboration of administrative rules to which activities within the area will be subject respective, in accordance with the provisions of this and other applicable laws; The measures that entity41 may impose for the preservation and protection of protected natural areas will only be those established, according to the respective matters, in this Law, entity116, national waters, entity149, entity58, and other that are applicable.

Table 6

Paragraph from the LAN law with negative charge for the entity in Spanish and its translation in English

"entity319": Acto emitido por el entity91 por causas de utilidad pública o interés público, mediante la declaratoria correspondiente, para extinguir: a. entity320 o asignaciones para la explotación, uso o aprovechamiento de entity321, de sus bienes públicos inherentes, ob. entity322, equipar, operar, conservar, mantener, rehabilitar y ampliar infraestructura hidráulica federal y la prestación de los servicios respectivos; XLVI. ', 'entity493 por causas de utilidad pública o interés público, declaratorias de rescate, en materia de concesiones para la explotación, uso o aprovechamiento de aguas nacionales, de sus bienes públicos inherentes, en los términos establecidos en la entity265; V. entity493 por causas de utilidad pública o

interés público, declaratorias de rescate de concesiones otorgadas por "la entity45", para construir, equipar, operar, conservar, mantener, rehabilitar y ampliar infraestructura hidráulica federal y la prestación de los servicios respectivos, mediante pago de la indemnización que pudiere corresponder; VI
"entity319 ": Act issued by the entity91 for reasons of public utility or public interest, through the corresponding declaration, to extinguish: a. entity320 or assignments for the exploitation, use or exploitation of entity321, of its inherent public goods, or b. entity322, equip, operate, conserve, maintain, rehabilitate and expand federal hydraulic infrastructure and the provision of the respective services; XLVI. ',' entity493 for reasons of public utility or public interest, declarations of rescue in the matter of concessions for the exploitation, use or use of national waters , of their inherent public goods, in the terms established in the entity265; V. entity493 for reasons of public utility or public interest, declarations of rescue of concessions granted by "the entity45", to build, equip, operate , conserve, maintain, rehabilitate and expand federal hydraulic infrastructure and the provision of services respective, by payment of the compensation that may correspond; VI ..

As it can be seen in Table 7, there are two sets of words for each of the paragraphs in which the entity "Aguas Nacionales" (National Waters) occurs, one of them made out from positive words and another from negative words. These sets of words were constructed from the results of the SentiStrength tool [21].

Table 7

Words that indicate the polarity at the inconsistency in Spanish and its translation in English

Law	**Words**	
	Positives	Negatives
LGEEPA	*realización*[+3] (realization)	
	preservación[+2] (preservation)	
	restauración[+2] (restoration)	
	protegidas[+2] (protected)	
	dispuesto[+2] (arranged)	
	medidas[+2] (measures)	
	protección[+2] (protection)	
	únicamente[+2] (only)	
LAN	*establecidos*[+3] (established)	
		explotación[-3] (explotation)
		servicios[-2] (services)

By using the WordReference dictionary of antonyms [19], it is noted that the word "explotación" (explotation) identified in the context of the entity "Aguas Nacionales" (National Waters), in the paragraph shown in Table 6 for the *LAN* law, has a relationship of antonymy with different words found in the paragraph of the *LGEEPA* law (Table 5) for the same entity; these words are shown in Table 8.

Table 8
Identified antonyms from the words that indicate the polarity in Spanish and its translation in English

Word (LAN)	Antonyms (LGEEPA)
explotación (exploitation)	*Realización*(realization)
	Preservación (preservation)
	protegidas (protected)
	protección (protection)

According to the methodology developed in the present work, it is concluded that the entity "Aguas Nacionales" (National Waters) presents a low level of coherence, given that when performing the sentiment analysis in the paragraphs in which this entity is mentioned in the *LGEEPA* and *LAN* laws it was identified that some of the words, that give polarity to the paragraph and that are related to this entity, are antonyms.

5.4 Opposite Polarity without Opposition Relation: Entity "Áreas" (Areas)

It was identified that the entity "Areas" appears in the LGEEPA and in the law "Ley General de Vida Silvestre" *LGVS* (General Wildlife Law); their paragraphs of appearance are shown in Tables 9 and 10 respectively. Based on the entity "Áreas" (areas), opposite polarities in the texts were obtained; nevertheless, when analyzing the vocabulary found in the context of this entity, there was no opposition relation obtained.

As it can be seen in Table 11, there are two sets of words for each one of the paragraphs in which the entity "Áreas" (Areas) occurs, one of them of positive words and the other of negative words. These sets of words were constructed from the results of the polarity analysis. Depending on the occurrences of these words a polarity was assigned to the paragraph, resulting in "Áreas" (Areas) associated with positive polarity for the *LGEEPA* law and negative polarity for *LGVS* law.

Table 9
Paragraph from the *LGEEPA* with positive charge for the entity "Áreas" (Areas) in Spanish and its translation in English

Se consideran áreas naturales protegidas: I. entity173; II. Se deroga. III. entity146 nacionales; IV. entity138 naturales; V. Se deroga. VI. áreas de protección de recursos naturales; VII. áreas de protección de flora y fauna; VIII. entity179; IX. entity147, así como las demás categorías que establezcan las legislaciones locales; X. entity224, así como las demás categorías que establezcan las legislaciones locales, y XI. áreas destinadas voluntariamente a la conservación. Para efectos de lo establecido en el presente Capítulo, son de competencia de la entity56 las áreas naturales protegidas comprendidas en las fracciones I a VIII y XI anteriormente señaladas. entity125 y del entity36, en los términos que señale la legislación local en la materia, podrán establecer parques, reservas

estatales y demás categorías de manejo que establezca la legislación local en la materia, ya sea que reúnan alguna de las características señaladas en las fracciones I a VIII y XI del presente artículo o que tengan características propias de acuerdo a las particularidades de cada entidad federativa. . .
The following are considered protected natural areas: I. entity173; II. It is repealed. III. entity146 nationals; IV. natural entity138; V. It is repealed. VI. Natural resource protection areas; VII. Flora and fauna protection areas; VIII. entity179; IX. entity147, as well as the other categories established by local laws; X. entity224, as well as the other categories established by local legislation, and XI. Areas voluntarily designated for conservation. For the purposes of what is established in this Chapter, the natural protected areas included in sections I to VIII and XI mentioned above are the responsibility of the entity56. entity125 and entity36, under the terms indicated by local legislation on the matter, may establish parks, state reserves and other management categories established by local legislation on the matter, whether they meet any of the characteristics indicated in sections I to VIII and XI of this article or that have their own characteristics according to the particularities of each federative entity.

Table 10
Paragraph from the *LGVS* with negative charge for the entity "Áreas" (Areas) in Spanish and its translation in English

... entity15 podrá establecer, mediante acuerdo entity48, hábitats críticos para la conservación de la vida silvestre, cuando se trate de: a) áreas específicas dentro de la superficie en la cual se distribuya una especie o población en riesgo al momento de ser listada, en las cuales se desarrollen procesos biológicos esenciales para su conservación. b) áreas específicas que debido a los procesos de deterioro han disminuido drásticamente su superficie, pero que aún albergan una significativa concentración de biodiversidad. c) áreas específicas en las que existe un ecosistema en riesgo de desaparecer, si siguen actuando los factores que lo han llevado a reducir su superficie histórica. d) áreas específicas en las que se desarrollen procesos biológicos esenciales, y existan especies sensibles a riesgos específicos, como cierto tipo de contaminación, ya sea física, química o acústica, o riesgo de colisiones con vehículos terrestres o acuáticos, que puedan llevar a afectar las poblaciones.
... entity15 may establish, through an entity48 agreement, critical habitats for the conservation of wildlife, in the case of: a) specific areas within the surface in which a species or population at risk is distributed at the time of listing, in the which essential biological processes are developed for their conservation. b) specific areas that due to deterioration processes have drastically decreased their surface area, but still harbor a significant concentration of biodiversity. c) specific areas in which there is an ecosystem at risk of disappearing, if the factors that have led it to reduce its historical surface continue to act. d) specific areas in which essential biological processes are developed, and there are species sensitive to specific risks, such as certain types of pollution, whether physical, chemical or acoustic, or risk of collisions with land or water vehicles, which may lead to damage populations..

Table 11
Words that indicate polarity in the context of the entity "Áreas" (Areas) in Spanish and its translation in English

Law	Words	
	Positives	Negatives
LGEEPA	*protegidas*[+2](protected) *protección*[+2] (protection) *voluntariamente*[+3] (voluntarily) *competencia*[+2] (competition) *acuerdo*[+2] (agreement)	
LGVS	*protección*[+2] (protection) *acuerdo*[+2] (agreement) *significativa*[+2] (significant) *sensibles*[3] (sensitive)	*críticos*[-2] (critical) *procesos*[-2] (processes) *riesgo*[-2] (risks) *deterioro*[-2] (deterioration) *riesgos*[-2] (risks) *contaminación*[-2] (pollution)

According to the results, in the *LGEEPA* law, the entity is in a context of protected areas, while in the *LGVS*, it is found in the context of critical habitats, so they get opposite polarities. However, at the antonym detection stage of the vocabulary obtained from the context of both paragraphs, there is no antonym relation in any pair of words that were found in the paragraphs from the *LGEEPA* and *LGVS* laws for this entity. Based on our methodology developed in the present work, it is concluded that the entity "Aguas Nacionales" (National Waters) has opposite polarities, but it is coherent by not presenting opposition relations.

5.5 Analysis of Results

The case study reflected in this document was obtained from the analysis of a set of eight laws, all of the ecological matter. It can be seen that even in documents focused on a particular subject, there can be identified opposition relations. By the analysis embodied in the Section 5.2, in Table 4, it is observed that the identified entities in different paragraphs present the same polarity in most cases. Only 2.3% of the entities are in paragraphs with opposite polarities. Those entities are considered for a deeper study because of the relation of those entities to their context. The example of the *"Aguas nacionales"* (National Waters) entity in Section 5.3 shows how the methodology, based on antonym relations in the context vocabulary, achieves the identification of a potential opposition relation. The *"Áreas"* (Areas) example (Section 5.4) highlights the importance of all the steps from this methodology (identify entities, obtain polarity and identify antonym relations) to mark an opposition relation since the polarity tested in the sentiment analysis is not enough for the marking. The consistency validation is carried out by experts in public policies, lawyers and legislators.

Conclusions

In this research, the concept of "opposition relations" in legal texts was defined, starting from antonymic relations in the vocabulary associated with entities whose paragraphs present opposite polarities when implementing sentiment analysis techniques.

The developed methodology consists of 4 stages, ranging from the computational transformation of texts to the coherence validation; going through two core stages: the identification of polarity in identified entities and the analysis of antonymic relations in the context. This methodology was implemented with computational algorithms and applied to the environmental scope of Mexican laws, but it is applicable in other areas.

The sentiment analysis tasks and the identification of entities were made with generic order tools. The polarity is evaluated using a dictionary, and for the marking of entities, a list of entities identified in news was used. Therefore, it is important that with the collaboration of different professionals, such as political scientists, lawyers and/or politicians, language resources could be generated, that allow characterization of the different concepts and relations that are involved in the diverse texts in the legal domain.

Acknowledgements

This work was assisted by the public politics team led by Dr. Harlan Koff, researcher from the “Instituto de Ecología A.C.” (INECOL) and from the University of Luxembourg. This collaboration is entitled in the project “Uso del big data para la gestión ambiental del desarrollo sostenible (Integralidad Gamma)”, FORDECYT-296842, under the direction of Dr. Miguel Equihua Zamora. The polarity analysis required specialized computational infrastructure and it was provided by the “Laboratorio Nacional de Supercómputo del Sureste de México” laboratory, thanks to the approved project 201903092N.

References

[1] H. Koff and C. Maganda: The EU and the human right to water and sanitation: Normative coherence as the key to transformative development. The European Journal of Development Research, 2016, Vol. 28, No. 1, pp. 91-110

[2] M. Rovalo Otero: La transversalidad del Derecho Ambiental: Un paradigma necesario en el Siglo XXI [The transversality of Environmental Law: A necessary paradigm in the 21st Century]. Política y Gestión Ambiental [Environmental Policy and Management], 2016, pp. 49-52

[3] Antena Radio: Diversidad Ambiental, Sección Medio Ambiente, ¿Qué puedo hacer yo? [Environmental Diversity, Environment Section, What can I do?] with Francisco Calderón Córdova / IMER - Horizonte 107.9 FM, - 1220 AM y - Radio México Internacional, 29/01/2018 [Online] [Last access: 02/02/2021] Available: http://diversidadambiental.org/medios/nota610.html

[4] E. L. Rissland, K. D. Ashley and R. Loui: AI and Law: A fruitful synergy. Artificial Intelligence, 2003, Vol. 150, No. 1-2, pp. 1-15

[5] R. Barros, A. Peres, F. Lorenzi and L. K. Wives: Case law analysis with machine learning in brazilian court. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 2018, Vols. 1 de 2 Springer, Cham, pp. 857- 868

[6] R. Giri, Y. Porwal, P. Chadha and R. Kaushal: Approaches for information retrieval in legal documents. 2017 Tenth International Conference on Contemporary Computing (IC3), 2017, Vol. IEEE, pp. 1-6

[7] Y. Chen, Y. Liu and W. Ho: A text mining approach to assist the general public in the retrieval of legal documents. Journal of the American Society for Information Science and Technology, 2013, Vol. 64, No. 2, pp. 280-290

[8] Y. Ma, P. Zhang and J. Ma: An ontology driven knowledge block summarization approach for Chinese judgment document classification. IEEE Access, 2018, Vol. 6, pp. 71327-71338

[9] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu and M. Sun: How Does NLP Benefit Legal System: A Summary of Legal Artificial, 2020, arXiv preprint arXiv:2004.12158

[10] M. Dragoni, S. Villata, W. Rizzi and G. Governatori: Combining NLP Approaches for Rule Extraction from Legal Documents. De 1st Workshop on MIning and RIasoning with Legal texts (MIREL 2016), 2016

[11] E. d. Maat, K. Krabben and R. Winkels: Machine Learning versus Knowledge Based Classification of Legal Texts. JURIX, 2010, pp. 87-96

[12] N. T. Son, N. T. P. Duyen, H. B. Quoc and N. L. Minh: Recognizing logical parts in Vietnamese legal texts using conditional random fields. The 2015 IEEE RIVF International Conference on Computing Communication Technologies-Research, Innovation, and Vision for Future (RIVF), IEEE, 2015

[13] M. Palmirani and R. Brighi: Metadata for the legal domain. 14th International Workshop on Database and Expert Systems Applications, 2003, Proceedings, 2003, Vol. IEEE, pp. 553-558

[14] K. D. Ashley and E. L. Rissland: Law, learning and representation. Artificial Intelligence, 2003, Vol. 150, Nos. 1-2, pp. 17-58

[15] K. H. Law, G. Lau, S. Kerrigan and J. A. Ekstrom: REGNET: Regulatory information management, compliance and analysis. Government Information Quarterly, 2014, Vol. 31, pp. S37-S48

[16] G. Lau, K. H. Law and G. Wiederhold: A Relatedness Analysis of Government Regulations using Domain, Knowledge and Structural Organization. Information Retrieval, 2006, Vol. 9

[17] X. Schmitt, S. Kubler, J. Robert, M. Papadakis and Y. LeTraon: A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate. 2019 Sixth International Conference on Social Networks Analysis, Management and Security, 2019 (SNAMS) IEEE

[18] ExplosionAI: Spacy, industrial-strength natural language processing. [Online], [Last access: 01/04/2021] Available: https://spacy.io/

[19] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede: Lexicon-based methods for sentiment analysis. Computational linguistics, 2011, Vol. 37, No. 2, pp. 267-307

[20] S. M. Jiménez Zafra, E. Martínez Cámara, M. T. Martín Valdivia and M. D. Molina González: Tratamiento de la Negación en el Análisis de Opiniones en Espanol [Treatment of Negation in Spanish Opinion Analysis], Procesamiento del Lenguaje Natural [Natural Language Processing] 2015, No. 54, pp. 37-44

[21] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai and A. Kappas: Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology, 2010, Vol. 61, No. 12, pp. 2544-2558

[22] M. Thelwall: Heart and soul: Sentiment strength detection in the social web with sentistrength. Cyberemotions: Collective emotions in cyberspace, 2014

[23] T. Baviera: Técnicas para el análisis del sentimiento en Twitter: Aprendizaje Automático Supervisado y SentiStrength [Techniques for Sentiment Analysis on Twitter: Supervised Machine Learning and SentiStrength], Dígitos, 2017, Vol. 1, No. 3, pp. 33-50

[24] UPIITA-IPN: Primer Coloquio Interdisciplinario para el Análisis de Legislación Ambiental [First Interdisciplinary Colloquium for the Analysis of Environmental Legislation], UPIITA-IPN, 13-14 08 2019 [Online] [Last access: 15/03/2021. Available: https://www.upiita.ipn.mx/novedades/coloquio-legislacionambiental

[25] C. Mattmann, Zitting J.: Tika in Action. Greenwich, CT., 2011

[26] J. D. Choi, J. Tetreault and A. Stent: It depends: Dependency parser comparison using a web-based evaluation tool. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015, (Volume 1: Long Papers)

[27] B. Landesman: WordReference.com, Reference Reviews, 2002

[28] H. Dubossarsky, E. Grossman and D. Weinshall: Outta Control: Laws of Semantic Change and Inherent Biases in Word Representation Models. Proceedings of the 2017 conference on empirical methods in natural language processing, 2017, pp. 1136-1145

Survey of Fake News Datasets and Detection Methods in European and Asian Languages

Maaz Amjad[1], Sabur Butt[1], Alisa Zhila[2], Grigori Sidorov[1], Liliana Chanona-Hernandez[3], and Alexander Gelbukh[1]

[1]Instituto Politécnico Nacional, Centro de Investigación en Computación (CIC), Gustavo A. Madero, 07738 Mexico City, Mexico, maazamjad@phystech.edu, sbutt2021@cic.ipn.mx, gelbukh@gelbukh.com

[2] Ronin Institute for Independent Scholarship, United States, alisa.zhila@ronininstitute.org

[3]Instituto Politécnico Nacional, ESIMEZ, lchanonah2100tmp@alumnoguinda.mx

Abstract: The presence of fake news and "alternative facts" across the web is a global phenomenon that received considerable attention in recent years. Several researchers have made substantial efforts to automatically identify fake news articles based on linguistic features and neural network-based methods. However, automatic classification via machine and deep learning techniques demands a significant amount of annotated data. While several state-of-the-art datasets for the English language are available and commonly utilized for research, fake news detection in low-resource languages gained less attention. This study surveys the publicly available datasets of fake news in low/medium-resourced Asian and European languages. We also highlight the vacuum of datasets and methods in these languages. Moreover, we summarize the proposed methods and the metrics used to evaluate the classifiers in identifying fake news. This study is helpful for analysis of the available sources in the lower resource languages to solve fake news detection challenges.

Keywords: datasets, fake news, low resource languages, deep learning, machine learning, evaluation metrics.

AMS Subject Classification: 68T50 Natural language processing, 68T01 General topics in artificial intelligence

1 Introduction

The fake news phenomenon imposes devastating and havoc impact worldwide. It poses not only technical challenges for social media platforms but also a dramatic impact on everyday life. Rampant "*online*" fake news leads to "*offline*" societal

events (e.g., the PizzaGate[1]). For example, according to the United Kingdom Office of National Statistics, anti-vaccination misinformation online reduced vaccination coverage across England and Wales[2]. In another example, Financial Times reported that French regulators had fined Bloomberg C5M for publishing a fake press release[3]. Therefore, social media platforms and other organizations should gear up to battle the dissemination of fake news and take preventive measures to maintain a trustworthy news ecosystem.

Manual verification of news articles is troublesome. Traditionally, journalists are required to verify claims against written or spoken facts. This requires a substantial amount of time and resources. For example, in PolitiFact[4] employs at least two news editors to authenticate the news article. Additionally, the amount of data is exploding, worldwide and in all languages, making detection of deceiving and spin information difficult because of its fast dissemination and easy availability. This brings the need for constant monitoring of digital content employing automatic fake news detection.

Automatic fake news detection is aimed to assist in monitoring and analyzing of giant amounts of data, and to reduce human efforts and time resources. Multiple advanced techniques have been investigated to approach fake news detection such as traditional (linear and non-linear) Machine Learning/Deep (ML/DL), Data Mining (DM), and Natural Language Processing (NLP). However, the most well-known research has been focused around the resource-rich languages, in terms of availability of tools, size of datasets, and previous research, predominantly Western, such as English [1,2,3].

In this paper, we survey the available resources for fake news detection from the perspective of Asian and European lower-resource languages. First of all, we want to derive attention to the size of the fake news problem in the regions where millions speak a variety of low to medium-resource languages of people. Next, we show that substantial effort exists for these languages to solve the fake news problem. We also gave a systematic comparison of fake news definitions used in various studies. Further, we provide a detailed analysis to highlight the points, where more improvement or effort is needed to achieve more impactful results.

Our contributions can be summarized as follows:

- We provide the first review of recent studies in low and medium resources, particularly, Asian and European languages for automatic fake news detection;

1 https://www.rollingstone.com/feature/anatomy-of-a-fake-news-scandal-125877

2 https://www.scl.org/articles/12022-the-real-world-effects-of-fake-news-and-how-to-quantify-them

3 https://www.ft.com/content/b082851a-07c1-11ea-a984-fbbacad9e7dd

4 https://www.politifact.com/

- We categorize and summarize publicly available datasets in Asian and European languages;
- We study and compare the various definitions of "fake news" used in the surveyed works;
- We identify different general approaches to fake news detection and group the studies accordingly;
- We overview the metrics used for fake news detection evaluation in the surveyed studies and show to which extent the results can be compared across different works;
- Finally, we identify the main challenges around the fake news detection problem and highlight the promising pathways for further research to solve fake news detection problem in Asian and European languages.

We hope this work may serve as a useful reference for the sources available to develop fake news detection systems for low-resource languages.

The remaining paper is structured as follows. Section 2 presents and discusses various definitions of the term fake news. Section 3 describes and groups the datasets. Section 4 sheds light on experimental methodologies employed in the development of fake news detection systems. It also presents and compares the results. Further, Section 5 provides comparison of popular evaluation metrics. Finally, Section 6 concludes with a discussion and Section 7 outlines future opportunities.

2 Variations of Definitions of Fake News

Multiple definitions of fake news were proposed in [1,4,5]. In the study [4], the authors presented several definitions of *disinformation* elaborated by multiple researchers (in contrast to *misinformation*). The study [4] concluded that disinformation has a specific goal, which is to provide information that misleads the reader.

In a similar study, researchers were investigating the ways the term "Fake news" was used [5]. The researchers categorized fake news into six types of news: fabrication, news satire, manipulation (e.g., editing pictures), advertising (e.g., ads depict as professional journalism), propaganda, and news parody. In the previous study [5], the scientists highlighted two popular themes among six types of news: the appropriateness and purpose of news articles.

The term "Fake news" has been defined from different perspectives. For example, satire can be defined as a news article that contains factually incorrect information. Nonetheless, the goal of this news article is not to deceive a reader by providing

unproven information but to highlight shameful, unethical, or otherwise "bad" attitudes. Finally, this brings up a new challenge to identify fake news because addressing this task demands clear definitions and examples to combat fake news on web-scale.

The year 2016 has been known as a *"post-truth"* era since it introduced recent advancements into traditional politics. In that view, Oxford Dictionary[5] announced *"post-truth"* as the word of the year 2016 shows that the sensitivity of fake news is a global problem. Similarly, Cambridge dictionary[6] called a news article fake news if it is propagated on the internet at a large scale to either use it as a joke or to influence public political ideologies.

Furthermore, in study [1], the authors classified fake news into three groups: serious fabrications, large scale hoaxes, and humorous fakes. The authors failed to provide specific reasons for using only these three categories. However, they shed light on the characteristics of each category and how to differentiate these three categories from each other. The same study also highlighted the limitations of datasets to perform fake news detection task. In addition to this, there is another type of fake news that is known as "clickbait", where the intent is to attract a consumer to click on a given link.

We propose the definition of fake news and fake news detection based on the previous works and analysis to define this term as follows:

- **Fake News:** Fake news is a factually incorrect news article and provides misleading information with the intent to deceive the readers making them believe it is true.
- **Fake News Detection:** For a given news article (unannotated) α, where $\alpha \in N$ (α is one news article out of N news article), an automatic fake news detection algorithm assigns score $S(\alpha) \in [0, 1]$ indicating the extent to which $S(\alpha)$ is assumed to be a fake news article.

For instance, if $S(\hat{\alpha})$ ¿ $S(\alpha)$, then it implies that $\hat{\alpha}$ has a higher tendency to be a fake news b b article. A threshold γ can be defined such that the prediction function F : $N \rightarrow$ [fake, not fake] is:

$$F(N) = \begin{cases} fake, & if \ \ S(\alpha) \in \gamma, \\ not\ fake, & \text{otherwise} \end{cases}$$

[5] https://en.oxforddictionaries.com/word-of-the-year/word-ofthe- year-2016

[6] https://dictionary.cambridge.org/us/dictionary/english/fake-news

3 Overview and Grouping of Fake News Datasets

In the era of artificial intelligence, data are essential assets to automate various computer-based tasks. In this view, automatic fake news detection is a striking but new area for the research community. However, fake news is a worldwide phenomenon and appears in all countries and in multiple languages. Several studies focusing on the English language achieved significant advancement and produced a few benchmark datasets in English.

Nevertheless, only limited sources in the form of datasets are available for poor resource-languages due to various reasons. The term "Fake news" is divided into many subcategories; this is why most publicly available datasets differ from one another and cannot be re-used in research with a slightly different focus within the broader "fake news" domain, as discussed in Introduction. Thirdly, data collection and annotation is a time-consuming and expensive task. Therefore, it is challenging to design new annotated datasets for fake news detection.

In this study, we primarily focus on non-English datasets available for automated fake news identification. Moreover, the inadequacy of fake news datasets is a major stumbling block, especially in automatically identifying fake news across multiple languages. We analyzed various datasets in related works that focused on assessing the integrity of news articles, Twitter postings, and YouTube comments.

We categorized the datasets into two sub-categories, (i) mainstream media articles datasets, (ii) social network posts datasets.

3.1 Mainstream Media Articles Datasets

Mainstream media articles datasets comprise on lengthy texts and news articles that can be seen in traditional e-newspapers, containing approximately 400 to 700 words. Below we describe three such datasets. The details of these datasets are presented in Tables 3 and 4.

3.1.1 Bend the Truth

In the recent study [6], a fake news dataset "Bend the Truth" is presented in the Urdu language that contains two types of news, (i) real news and (ii) fake news. The dataset covered five news topics, sports, entertainment, business, technology, and health.

The authors collected 500 real news from five different domains. Each domain is contributing 100 news in the proposed corpus. Since there are five categories of news, so in total, the dataset contains 500 real news. The real news in the Urdu language was manually collected from 16 news stream websites and four different countries from January 2018 to December 2018. These countries are the USA, UK, India, and Pakistan. The study provides a comprehensive description of real news collection and annotation methodology.

3.1.2 BanFakeNews

Study [7] introduces the fake news dataset BanFakeNews in Bangla language. Bangla language is the official language in Bangladesh and has more than 230 million native speakers and is spoken widely in Bangladesh and India.

The dataset contains three categories of news, (i) click baits, (ii) satirical, (iii) real, and fake news with their headlines. The dataset considered real and fake news in one category.

The dataset includes 242 topics that were further classified based on similar categories into 12 news domains (sports, politics, crime, technology, etc.). 48,678 real news were collected (till March 08, 2020) from 22 mainstream news websites in Bangladesh, each real news on average had 271.16 words. In contrast, only 1,299 fake news articles were collected from www.jaachai.com and www.bdfactcheck.com, and each fake news contained, on average, 276.36 words. The meta-data such as the source of the news article, publication time, news topic, and the relation between headline and article were provided for only 8500 news articles due to the task complexity. The assessment or labelling was done by undergraduate students who have a background in Computer Science and Engineering and Software Engineering who manually annotated the source of the news and tagged the relation between headline and article as "related" or "non-related."

3.1.3 Persian Stance Dataset

A recent study [8] contributed the first stance detection dataset in the Persian language to study article-claim stance and headline-claim stance classification tasks. The study created a web-based tool (the stance detection system)[7] to collect claims and news articles. Stance detection is defined as investigating what other mainstream news organizations publish about a piece of news, i.e., understanding what these organizations claim about that specific news [9] and on Twitter [10]. The authors defined a claim as news published by another news agency, and that claim was used to check the stance of the body of the news article. The same web-based tool was used to annotate the news article's stance against the claim made and to find the integrity of each claim. All the claims were collected and created from rumors and news headlines using two Iranian websites (Fakenews and Shayeaat).

The study provides 2,124 news articles and textual claims (news headlines) to the stance detection system that annotates the news article's stance against the claim into four groups, agree, disagree, discuss, and unrelated. The research reported that Fakenews and Shayeaat assemble rumors (headlines) from different sources and manually check the news articles' credibility.

[7] https://github.com/majidzarharan/persian-stanceclassification

3.1.4 Arabic Fake Rumours Dataset

A recent study [11] analyzed fake news in the context of rumor detection. The study presented a corpus in the Arabic language for the automatic fake rumor detection task. It considered rumor detection as a binary classification problem. The authors focused on three Arab celebrities, Fifi Abdu (an Egyptian dancer), Abdelaziz Bouteflika (the former Algerian president), and Adel Imam (an Egyptian comedian).

To retrieve the YouTube comments, YouTube API was used to collect the comments associated with these three personalities' death rumors. The authors considered YouTube comments as potential rumors to address automatic fake new detection tasks. Furthermore, it is essential to mention that the term "fake news" authors didn't mean "news articles".

The researchers used keywords associated with rumors to data-mine fake stories (comments) related to these personalities' death. For example, the study used keywords such as "*Algerian president dies*", "*yes death*," "*Bouteflika death*," to retrieve rumors related to the death of the Algerian president. Similarly, they mined comments associated with the death of an Egyptian comedian using the keywords "*adel imam dies*", "*Adel die*," and "*Allah yerhamo*." Likewise, the Egyptian dancer's death rumours were collected using "*Fifi died*", "*True news*", and "*Allah yarhemak*". If one of these keywords appeared in a comment, then the comment was tagged as a rumor. In the contrary case, the comment was labelled as the no-rumor if the comment contained the celebrity's name and did not mention death. In the end, 343 rumors and 3092 non-rumors were included in the final detest.

3.1.5 Czech, Polish, and Slovak Fact-checking Datasets

The study [12] presented datasets in three languages: Czech, Polish, and Slovak to address fact-checking tasks in West Slavic Languages. The Czech dataset contained 9082 claims of politicians that were annotated by expert annotators in four classes: (i) False, (ii) True, (iii) Unverifiable, and (iv) Misleading. Likewise, the dataset also contained 2835 politicians claims in Polish and 12554 politicians claims in Slovak language. However, the authors did not mention whether the claims of the same politicians were used in three languages. The authors downloaded the claims from websites in April 2018.

3.1.6 DANFEVER (Danish) Dataset

A new dataset in Danish has been proposed [13] for the claim detection task. The dataset contained 6,407 claims in Danish language that are manually annotated into three classes: (i) Supported claims, (ii) Refuted, and (iii) NotEnoughInfo claims. Different sources, such as Danish Wikipedia and Den Store Danske have been used for claims generation.

3.1.7 FactCorp (Dutch) Dataset

The author in [14] proposes to investigate fact-checks from a corpus linguistic approach. This study aims to understand and learn more about the extent and substance of factchecks, additionally more about that how science appears (incorrectly) in the news and how to behave from the science communication perspective. A FactCorp contains a 116 million words corpus and 1974 fact-checks reported from three different Dutch newspapers. The author of this study has done different analyses as a result, including keyword, qualitative content elements, and rhetorical moves analysis. According to these analyses, they show that FactCorp allows a wealth of possible applications, emphasizing the need to develop such resources.

In the study [15], the researcher argues that network analysis's persistent disregard for conflict leads to enormous conclusions on heated arguments. The researcher in this study introduces a method for incorporating negative user-to-user contact into online arguments by analyzing signed networks with negative and positive relationships. The 'black Pete' debate on Twitter is analyzed on the annual Dutch celebration in this study. The dataset containing 430,000 tweets is used, and ML and NLP-based solutions are applied to identify the stance of users in online debate and the interaction between users. The results demonstrate that some groups are targeting each other, while others appear to be scattered across Twitter.

3.1.8 DEAP-FAKED (Estonian)

Recently, hoaxes and fake news spreading on social media have attracted more attention, especially in politics and healthcare (COIVD-19). For the detection of fake news on social media platforms, a Deep-Facked framework has been proposed in [16]. A deep-Facked approach is the combination of NLP-based and GNN-based techniques. Two different publicly available databases containing articles from the healthcare, politics, business, and technology domain are used in the Deep-Facked approach.

3.1.9 Cresci-2017 (Finnish)

The goal of this study is to investigate the influence of bots on Finnish politics Twitter, using a dataset of accounts that follow important Finnish politicians before the 2019 parliamentary election.

In this social media life, opinion mining and sentiment analyses are important tasks, e.g. when stipulating fake and hoax news. In this study, the author [17] addressed this deficiency by presenting a 27,000-sentence data set that was annotated with sentiment polarity by three native annotators separately. They used the same three annotators throughout the data set, which gives the unique opportunity to study annotator behavior across time. Furthermore, they examine their inter-annotator agreement and present two baselines to verify the utility of the dataset.

A new dataset in the Finnish language on rumor detection is presented [18]. In this study, the author assesses two different models based on LSTM and two models based on BERT. Because the models were trained on tiny and biased corpora, these findings suggest that additional work is needed for pre-trained models in the Finnish language.

3.1.10 Fake.Br Corpus (Portuguese)

The study [19] proposed a news corpus for fake news detection in Brazilian Portuguese (PT). The dataset contained 7,200 news, which was manually labelled and contained an equal number of fake news (3,600) and true news (3,600) articles. The news articles were retrieved from January of 2016-2018.

3.1.11 Partelet Corpus of Propaganda Texts (Hungarian)

A digital Hungarian language database of communist propaganda text named as Partelet has been presented in [20]. This paper serves two purposes: first, to provide a general overview of the corpus compilation method and basic statistical data, and second, to demonstrate the dataset utility using two case studies. Results illustrate that the proposed corpus offers a unique potential for doing research on Hungarian propaganda speech as well as assessing changes in this language using computer-assisted approaches over 35 years.

Recent developments in the field of semantic encoding demonstrate significant progress and call attention to such strategies. These approaches' main purpose is to convert human-written natural language text into a semantic vector space. The train and execution of a semantic encoder for the Hungarian language are discussed in this study [21]. Since Hungarian is not a commonly spoken language, the number of linguistic available resources is restricted. Although the method described here is used with the Hungarian language, it may be used in any small or medium-sized language.

3.1.12 Spanish Fake News Corpus (Spanish)

The Study [22] introduced the first Spanish corpus to investigate and analyze the style-based fake news detection in the Spanish language. The dataset included an overlap of distinct news topics and classes containing true news (491) and fake news (480). The news was manually tagged and obtained from January to July of 2018 from several websites.

3.1.13 Fake News Polarization (Italian)

The study [23] aims at disseminating fake news on Facebook pages. The dataset consisted of 333,547 news officials and 51,535 fake news on Facebook posts which were further divided into "entities" (i.e., news topics). The data was collected in July-December 2016 exclusively by means of the Facebook Graph API.

3.1.14 CT-FAN-21 Corpus (Bulgarian, Turkish, Spanish)

The research [24] investigated into misleading news articles in European languages including Spanish, Turkish and Bulgarian. They tested out their CT-FAN-21 corpus on 900 trained and 354 test articles submitted by 27 teams for Task 3A, 20 teams for task 3B assigned for 1) 3A; topical domain detection of news articles and 2) 3B; multi-class fake news detection.

3.1.15 FakeDeS: Spanish dataset for Fake news

Datasets for fake news in Spanish are available though not in abundance [22,25,26]. In 2021 IberLeF released [25] the second iteration of the fake news challenge named "FakeDeS". The first edition was released [26] in 2020 named as "MEX-A3T". The second edition of the dataset used "MEX-A3T" dataset as the training set and created a new test dataset with data related to COVID-19. The topic distribution of the dataset comprised of science, society, health, politics, entertainment, education, economy and sport.

The dataset was compiled using fact-checking websites and newspapers. The second edition of the dataset has 970 (491 True, 480 Fake) training files and 572 (286 True, 286 Fake) test files, while the first edition contains 676 (338 True, 338 Fake) training files and 295 (153 True, 142 Fake) test files.

3.1.16 Fake News Dataset for Slovak

A dataset in the Slovak language is presented [27] with a focus on home news, world news, and economic news. However, in this paper, we discuss the extension of this dataset introduced in the paper [28] with deep learning baselines. The data was obtained from multiple news sources targeted at a specific domain of Slovak home news. The targeted news was annotated with labels 0 (Fake News) or 1 (True News) using konspiratori.sk (database for news credibility) at the initial stage and then manually verifying it. The final distribution shaped into 11,410 (training), 3,803 (validation), and 3,804 (test) articles respectively.

Table 1
Mainstream / Social Media Articles Datasets in Asian languages.

		Fake News Datasets			
Name	Language	Size	Main Input	Task	Annotation
Bend the Truth	Urdu	Real: 500 Fake: 400	News Articles	Fake News Detection	Professional Journalists
BanFake News	Bangla	Real: 48678 Fake: 1299	News Articles	Fake News Detection	Trained annotators
Persian Stance	Persian	Articles: 2124 Claims: 600	News Articles	Stance Detection	Trained annotators
Arabic Rumours	Arabic	Rumours: 343 Non-rumours: 3092	YouTube Comments	Fake Rumours Detection	Trained annotators

4 Comparison of Methods for Fake News Detection

Several studies have been conducted to understand and investigate multiple ways to automatically differentiate fake news from real news. This study analyzes the essential research in which we find work related to fake news detection tasks in Asia. Table 5 shows the proposed techniques in low-resource Asian languages. We limited the scope of this study by only analyzing the methods used in Asian languages, and would like to work on methods for European Languages in our future work. To the best of our knowledge, no prior research has been done to analyze automatic fake news detection systems in lowresource Asian languages. We categorize them into two subsections: Non-Neural Network Techniques and Neural Network Techniques.

4.1 Features for Fake News Detection

There are two main methods to tackle the fake news detection task (i) analyzing the content of the news article and (ii) analyzing the context of the news article. In the first method, a recent study comprehends the fake news detection phenomenon; it reveals that fake news tends to spread faster than real news [2].

In contrast, in the second method, linguistic features differentiate fake news articles from real news articles, i.e., discussing typical patterns. For example, in recent studies, linguistic features have been used to perform automatic fake news identification task [3, 6-8, 11, 29]. It is essential to highlight that most of the studies on fake news detection lack concrete guidelines on what features are necessary for the task. This is significant to know because these studies use specific data and feature sets to train classifiers. Moreover, the studies also lack details about why fake news is classified as fake news and the classifiers' decision behind classifying fake news articles.

Table 2
Mainstream / Social Media Articles Datasets in European languages

Fake News Datasets					
Name	Language	Size	Main Input	Task	Annotation
Czech fact-checking	Czech	True: 5669 False: 1222 Unverifiable: 1343 Misleading: 848	Claims of politicians	Fact-Checking Detection	Trained annotators
Polish fact-checking	Polish	True: 1761 False: 648 Unverifiable: 113 Misleading: 313	Claims of politicians	Fact-Checking Detection	Trained annotators
Slovak fact-checking	Slovak	True: 7987 False: 1670 Unverifiable: 1751 Misleading: 1146	Claims of politicians	Fact-Checking Detection	Trained annotators
	Slovak	True: 9979	News articles	Fake news	

Slovak Fake news		Fake: 9048		Detection	konspiratori.sk annotators
DANFEVER	Danish	Supported Claims: 3,124 Refuted Claims: 2,156 Notenoughinfo Claims: 1,127	Text annotated as claims	Claim Verification	Trained annotators
FactCorp	Dutch	Fact-Checks: 1,974	Dutch news	Fact-Checking	Trained annotators
Deep-Faked	Estonian	True: 9,129 Fake: 5,058	News article	Fake News Detection	Trained annotators
Cresci-2017	Finnish	Bots Account: 3000 Genuine Account: 3000	Claims of politicians	Identify the bot account	Trained annotators
Suomi24	Finnish	Sentences: 27,000	Social website	Fake news detection	Trained annotators
Fake.Br Corpus	Portuguese	True: 3600 Fake: 3600	News article	Fake News Detection	Trained annotators
Spanish Fake News Corpus	Spanish	True: 491 Fake: 480	News website	Fake News Detection	Trained annotators
FakeDeS	Spanish	True: 777 Fake: 766	News articles Fact-Checking Websites	Fake news Detection	Trained annotators
Fake News Polarization	Italian	Official: 333,547 Fake: 51,535	Facebook	Fake News Detection	Trained annotators
Partelet	Hungarian	Text Tokens: 13,185,200	Partelet journel	Propaganda Detection	Trained annotators
HoaxItlay	Italian	News: 37k	Twitter streaming API	Fact-checking/Disinformation	Trained annotators
CT-FAN-21 Corpus	Bulgarian, Turkish, Spanish	False:111 True: 65 Partially False: 138 Others: 40	News Articles	Fact-Checking Detection	Trained annotators

4.2 Non-Neural Network Techniques

Most studies used linguistic features to adders to the automatic fake news detection task. Researchers have been using linguistic features such as N-grams, syntactic features such as POS tags, and semantic features like text entailment and metadata (the headline's lengths and the body of news articles) to implement fake news classification on the benchmark datasets.

A recent study [11] presented a fake news corpus in the Arabic language. The study focused on fake news detection in their dataset using three machine learning classifiers. The experiments were performed with the train test split ratio 70/30, respectively using N-grams features, namely, word N-grams where N varies from uni-gram to tri-gram, with term frequency-inverse document frequency (TF-IDF) weighting scheme. Three supervised machine learning algorithms have been used, such as Decision Tree (DT), Support Vector Machine (SVM) with linear kernel, and

Multinomial Naıve Bayes (MNB) classifiers. The study reported that the SVM achieved the highest accuracy of 0.95 compared to other classifiers in classifying rumors in YouTube comments.

A similar study [6] on fake news detection in Bangla language, used linguistic features such as word N-grams (n=1,2,3) and character N-grams (N=3,4,5) along with the normalized frequency of different POS tags. The study removed stop words and punctuation in the pre-processing phase. Additionally, the research utilized metadata (the headline's lengths and the body of news articles) and punctuation frequency as features. Furthermore, to convert words into vectors, the study used TFIDF as the frequency weighting scheme. For the classification, linguistic features were supplied into a linear Support Vector Machine (SVM), Random Forest (RF), and a Logistic Regression (LR) model. For the experiments, the split data ratio was 70/30 train-to-test, respectively. The SVM model outperformed other classifiers and achieved 0.89 F1-score and 0.90 F1-score using character 3-gram weighted frequencies and all linguistic features.

Likewise, research [7] investigated automatic fake news detection in the Urdu language. The study classified news articles using combinations of different N-gram types (words, characters, and functional words). It showed that the combinations provide better results than N-grams of a single type. The experiments used five N-gram frequency weighting schemes (TFIDF, normalized, log-entropy, binary, TF) and seven different machine learning classifiers. The study provided a comprehensive analysis of different feature sets used in the experiments. Lexical features with N-gram size 1 to 3 obtained better results compared with 4,5,6. Finally, the study reported that AdaBoost outperformed other classifiers by getting 0.86 F1-real and 0.90 F1-fake scores. The authors also reported a balanced accuracy of 0.88.

Previous study [8] on stance classification in the Persian language used three machine learning classifiers. The study reported that two feature types, such as bag-of-words representation (BoW) and TFIDF, were used. Eventually, the study showed that Random Forest achieves an accuracy of 0.69 in recognizing the stance of headline-claim.

Study [30] used term frequency (TF) weighting scheme and Naive Bayes classifier. The study reported that Native Bayes obtained 0.78 accuracy to identify hoax news using the Indonesian language.

4.3 Neural Network Techniques

In recent studies, Deep Learning techniques have been widely used in different tasks such as text classification and generation tasks. These techniques, namely, Neural Networks, achieved significant results and showed impressive performance in solving various NLP-related tasks. Different neural network architectures such as the Convolutional Neural Network, Recurrent Neural Network, and Transformer all

need much data to learn hidden patterns. These techniques obtain better results than linguistic feature-based methods.

The study [6] used semantic features to differentiate fake articles from real news articles. The experiments were conducted based on two types of word embeddings (vector representations of each news article) Fast text word embeddings [31] (300-dimensional word vectors) and Word2Vec [32] (100dimensional word vectors). The research used 256 different kernels, having to vary in size lengths from 1 to 4. The global max pool and the average pool were used in the pooling layer. For the activation function, ReLU [18] activation function was used.

In the prior study [8], the study focused on both tasks, headline-claim stance classification and articleclaim stance classification. The research was based on deep learning techniques, particularly the stack LSTM architecture using pre-trained 300-dimensional word embedding. All the experiments were performed with the deep learning library Keras[8]. 100-word embedding features are fed to two LSTMs to consider word sequences. The neural network has three dense layers, and each layer contained 300 neurons. In the last layer of the neural network, the softmax activation function is used to obtain the final output. The headline text was fed as input to the neural network. In addition to this, the study investigated two tasks, headline-claim stance detection, and article-claim stance detection. Thus, the authors reported that stackLSTM did not perform well in recognizing the headline-claim (in this task, the Random Forest classifier outperformed deep learning). However, the study illustrated that stackLSTM exceeded other techniques by obtaining 0.72 accuracy in finding the article-claim stance.

We observed that Deep Learning methods are not so prominent in addressing the automatic fake news detection task, especially in European and Asian low-resource languages, for several reasons.

First of all, the inadequacy of available sources in the form of datasets. Secondly, creating datasets is a time-taking task, but it requires financial support, which is a challenging part most of the time. Thirdly, the research community in Asian and European low-resource languages is minimal. Finally, the available datasets are small in size, therefore, not sufficient to train Deep Learning techniques to tackle automatically identifying fake news. Table 3 and Table 4 show the size of the available datasets in Asian and European languages for differentiating fake news.

5 Popular Evaluation Metrics

This section explains various evaluation approaches and metrics used to assess the performance of different fake news detection systems. Fake News detection is

[8] https://keras.io

almost universally approached as a classification task, either as a binary classification (more often) or a multi-label task. A binary classification problem has the goal to classify the instances of a given set into two categories. For example, in fake news detection as a binary classification task, the goal is to differentiate fake news from real news. However, if a problem is concerned with more than two groups, it is a multi-class task. For example, detecting a stance between a news headline and the whole text of the news article is a multi-class task since there are more than two labels involved, such as *agree disagree*, *unrelated*, etc.

In general, studies on fake news detection used different metrics to evaluate the performance of the presented methods, creating some inconvenience compared to the results among different works. For example, we observe studies reporting precision, recall, accuracy, F1 score, and ***ROC-AUC*** to evaluate the performance of various models trained on balanced datasets. In contrast to the balanced datasets, multiple studies have reported precision, recall, along with the F1 score for the fake class and ROC-AUC to examine the overall system quality and evaluate the model performance. In addition to this, some studies also calculated ***Micro-F1*** scores for highly unbalanced datasets. Finally, we noticed that most of the studies used the F1 score to measure the model's performance since most of the datasets are unbalanced.

Discussion and Conclusions

This work provides the first overview of various publicly available datasets for automatic fake news detection in Asian and European languages. Most of which are poor resource languages, providing comparative statistics on their sizes, grouping them by the length and source of content news, and surveying dataset annotation procedures. We have also surveyed the approaches used in the studies on fake news detection in low-resource languages and grouped them into studies using traditional machine learning and neural network approaches. We note that working on Asian languages with more resources, notably, Chinese, demonstrates wider adoption of neural networks and achieves better results with those. Finally, we provide a brief overview of the evaluation metrics used to report fake news detection performance. It is important to note that due to a large variety of metrics available, some studies choose to report different metrics than others, which leads to difficulties in comparison among studies.

Although low-resource languages have limited resources and a plethora of challenges, these languages lack expert-based fact-checking websites, i.e., PolitiFact[9] or FactCheck[10], which provide the services of fact-checking. However, tackling fake news tasks in low-resource languages can decrease the detrimental consequences of fake news globally. Multiple studies reported fact extraction [33] and relation extraction [34] in English, but this research still needs attention in low resource languages. Unless these techniques are enhanced, robust Knowledge Bases

[9] https://www.politifact.com/

[10] https://www.factcheck.org/

(KB) cannot be created for fact-checking, to eliminate fundamental issues like redundancy [35], invalidity [36], conflicts [37], unreliability [38] and incompleteness [39,40] in fake news. Style detection in fake news identifies the intent of the content and the style of text changes across languages and domains. The textual style also evolves with time, and hence more attention should be put to create solutions needed for style-based fake news detection in low-resource languages. While targeting the fake news, it is also important to analyse the check worthiness [41] of the news, which can be analysed by the potential of influence [42], user reputation [43], historical likelihood of the topic and title verification [44] of the content. This improved the efficiency of fake news that can have a mass impact on society and unfortunately, most of these topics need emphasis by the research community of the low resource languages.

We also note that one of the crucial difficulties faced while assembling the fake news datasets was finding the datasets focusing solely on fake news detection. In many cases, datasets are purposed for multiple tasks that are only indirectly related to the fake news detection problem, particularly, the datasets annotated for rumor detection, stance detection, and differentiating between fake news sub-classes satire.

Future Opportunities and Research Directions

Future research on fake news detection might extend datasets' explanations in most major languages used in Natural Language Processing to study fake news from various perspectives. We also want to track attention to explainable machine learning algorithms to solve automatically fake news detection. The explainable algorithm can point out important features and the classifiers' decision behind classifying news articles as fake or not fake. This can significantly improve the performance of existing fake news detection systems. We also want to investigate the performance of the machine learning algorithm trained on one dataset and test it on a different dataset across languages. For example, we want to study whether training on stance detection dataset and testing on rumor detection can provide better performance. In future research, we also want to analyze the methods used in European languages.

Acknowledgment

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico and grants 20211784, 20211884, and 20211178 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico.

References

[1] Rubin, V., Chen, Y. & Conroy, N. Deception detection for news: three types of fakes. Proceed. of the Association for Information Science and Technology, V.25, N.1, 2015, pp.1-4.

[2] Lazer, D., Baum, M., Benkler, Y., Berinsky, A., Greenhill, K., Menczer, F., Metzger, M., Nyhan, B., Pennycook, G., Rothschild, D. & Schudson M. The science of fake news. Science. V.359, N.6380, 2018, pp.1094-1096.

[3] Perez-Rosas, V., Kleinberg, B., Lefevre, A. & Mihalcea, R. Automatic detection of fake news. ArXiv Preprint ArXiv:1708.07104, 2017.

[4] Fallis, D. A functional analysis of disinformation. Int. Conf. 2014 Proceedings, 2014, pp.621-627.

[5] Tandoc Jr, E., Lim, Z. & Ling, R. Defining "fake news" A typology of scholarly definitions. Digit. Journal., V.6, N.2, 2018, pp.137-153.

[6] Amjad, M., Sidorov, G., Zhila, A., Gómez-Adorno, H., Voronkov, I., & Gelbukh, A. Bent the truth: A benchmark dataset for fake news detection in Urdu and its evaluation. J. Intell. Fuzzy Syst., V.39, 2020, pp.2457-2469.

[7] Hossain, M., Rahman, M., Islam, M. & Kar, S. Banfakenews: A dataset for detecting fake news in Bangla. ArXiv Preprint ArXiv:2004.08789, 2020.

[8] Zarharan, M., Ahangar, S., Rezvaninejad, F., Bidhendi, M., Jalali, S., Eetemadi, S., Pilehvar, M. & Minaei-Bidgoli, B. Persian Stance Classification Dataset, 2019.

[9] Pomerleau, D. & Rao, D. The fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. Fake News Challenge, 2017.

[10] Sadr, M., Mousavi Chelak, A., Ziaei, S. & Tanha, J. A predictive model based on machine learning methods to recognize fake persian news on twitter. Int. J. Nonlinear Anal. Appl., V.11, 2020, pp.119-128.

[11] Alkhair, M., Meftouh, K., Smaïli, K., & Othman, N. An Arabic corpus of fake news: Collection, analysis and classification. Int. Conf. on Arabic Language Processing, 2019, pp.292-302.

[12] Přibáň, P., Hercig, T., & Steinberger, J. Machine learning approach to fact-checking in West Slavic languages. Proceed. of the Int. Conf. on Recent Advances in Natural Language Processing, 2019, pp.973-979.

[13] Nørregaard, J. & Derczynski, L. DANFEVER: Claim verification dataset for Danish. Proceed. of the 23rd Nordic Conference on Computational Linguistics, 2021, pp.422-428.

[14] Meulen, M. & Reijnierse, W. FactCorp: A Corpus of Dutch Fact-checks and its Multiple Usages. Proceedings Of The 12th Language Resources And Evaluation Conference, 2020, pp. 1286-1292

[15] Keuchenius, A., Tornberg, P. & Uitermark, J. Why it is important to consider negative ties when studying polarized¨ debates: A signed network analysis of a Dutch cultural controversy on Twitter, PloS One, 2021.

[16] Mayank, M., Sharma, S. & Sharma, R. DEAP-FAKED: Knowledge Graph based Approach for Fake News Detection. ArXiv Preprint ArXiv:2107.10648, 2021.

[17] Linden, K., Jauhiainen, T. & Hardwick, S. FinnSentiment–A Finnish Social Media Corpus for Sentiment Polarity Annotation. ArXiv Preprint ArXiv:2012.02613, 2020.

[18] Hämäläinen, M., Alnajjar, K., Partanen, N., & Rueter, J. Never guess what I heard... Rumor Detection in Finnish News: a Dataset and a Baseline.ArXiv Preprint ArXiv:2106.03389, 2021.

[19] Monteiro, R., Santos, R., Pardo, T., De Almeida, T., Ruiz, E. & Vale, O. Contributions to the study of fake news in Portuguese: New corpus and automatic detection results. Int.Conf. on Computational Processing of the Portuguese Language, 2018, pp.324-334.

[20] Kmetty, Z., Vincze, V., Demszky, D., Ring, O., Nagy, B. & Szabo, M. P´art´elet: A Hungarian corpus of propaganda´ texts from the Hungarian socialist era. Proceedings Of The 12th Language Resources And Evaluation Conference, 2020, pp. 2381-2388.

[21] Kantor, A., Kiss, A. & Grad-Gyenge, L. Semantic Encoder Tasks for the Hungarian.´

[22] Posadas-Durán, J., Gómez-Adorno, H., Sidorov, G. & Escobar, J. Detection of fake news in a new corpus for the Spanish language. J. Intell. Fuzzy Syst., V.36, N.5, 2019, pp.4869-4876.

[23] Vicario, M., Quattrociocchi, W., Scala, A. & Zollo, F. Polarization and fake news: Early warning of potential misinformation targets. ACM Transactions On The Web (TWEB), V.13, 2019, pp.1-22.

[24] Shahi, G., Struß, J. & Mandl, T. Overview of the CLEF-2021 CheckThat! lab task 3 on fake news detection. Working Notes Of CLEF, 2021.

[25] Gómez-Adorno, H., Posadas-Durán, J., Bel-Enguix, G. & Capetillo, C. Overview of FakeDeS at IberLEF 2021: Fake News´ Detection in Spanish Shared Task. Proces. Leng. Nat., V.67, 2021, pp.223-231.

[26] Aragón, M. E., Jarquín-Vásquez, H. J., Montes-y-Gómez, M., Escalante, H. J., Pineda, L. V., Gómez-Adorno, H., Posadas-Durán, J. & Bel-Enguix, G. Overview of MEX-A3T at IberLEF 2020: Fake news and aggressiveness analysis in Mexican Spanish. IberLEF@ SEPLN, 2020, pp.222-235.

[27] Sarnovsky, M., Maslej-Kresnova & Hrabovska, N. Annotated dataset for the fake news classification in Slovak language. 18th Int. Conf. on Emerging ELearning Technologies and Applications (ICETA), 2020, pp.574-579.

[28] Ivancová, K., Sarnovský, M., & Maslej-Krcšňáková, V. Fake news detection in Slovak language using deep learning techniques. 2021 IEEE 19th World

Symposium on Applied Machine Intelligence and Informatics (SAMI), 2021, pp.000255000260.

[29] Patwa, P., Bhardwaj, M., Guptha, V., Kumari, G., Sharma, S., Pykl, S., Das, A., Ekbal, A., Akhtar, M.S. & Chakraborty, T. Overview of constraint 2021 shared tasks: Detecting English covid-19 fake news and Hindi hostile posts. In International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation, Springer, Cham, 2021, pp. 42-53.

[30] Pratiwi, I., Asmara, R. & Rahutomo, F. Study of hoax news detection using naive bayes classifier in Indonesian language. 2017 11th Int. Conf. on ICTS, 2017, pp.73-78.

[31] Grave, E., Bojanowski, P., Gupta, P., Joulin, A. & Mikolov, T. Learning word vectors for 157 languages. ArXiv Preprint ArXiv:1802.06893, 2018.

[32] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process., 2013, pp.3111-3119.

[33] Thorne, J., Vlachos, A., Christodoulopoulos, C. & Mittal, A. Fever: A large-scale dataset for fact extraction and verification. ArXiv Preprint ArXiv:1803.05355, 2018.

[34] Yu, B., Zhang, Z., Liu, T., Wang, B., Li, S. & Li, Q. Beyond word attention: Using segment attention in neural relation extraction. IJCAI, 2019, pp.5401-5407.

[35] Altowim, Y., Kalashnikov, D. & Mehrotra, S. Progressive approach to relational entity resolution. Proceed. of the VLDB Endowment, V.7, N.11, 2014, pp.999-1010.

[36] Hoffart, J., Suchanek, F., Berberich, K. & Weikum, G. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artif. Intell., V.194, 2013, pp.28-61.

[37] Kang, B. & Deng, Y. The maximum Deng entropy. IEEE Access, V.7, 2019, pp.120758-120765.

[38] Ye, J. & Skiena, S. Mediarank: Computational ranking of online news sources. Proceed. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining, 2019, pp.2469-2477.

[39] Kazemi, S. & Poole, D. Simple embedding for link prediction in knowledge graphs. ArXiv Preprint ArXiv:1802.04868, 2018.

[40] Shi, B. & Weninger, T. Open-world knowledge graph completion. Proceed. of the AAAI Conf. on Artif. Intell., V.32, 2018.

[41] Hassan, N., Arslan, F., Li, C. & Tremayne, M. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. Proceed. of

the 23rd ACM SIGKDD Int. Conf. on Data. Min. Knowl. Discov., 2017, pp.1803-1812.

[42] Vosoughi, S., Roy, D. & Aral, S. The spread of true and false news online. Science, V.359, N.6380, 2018, pp.1146-1151.

[43] Ferrara, E., Varol, O., Davis, C., Menczer, F. & Flammini, A. The rise of social bots. Commun. ACM., V.59, N.7, 2016, pp.96-104.

[44] Ghanem, B., Rosso, P. & Rangel, F. Stance detection in fake news a combined feature representation. Proceed. of the First Workshop on Fact Extraction And VERification (FEVER), 2018, pp.66-71.

A Review and Perspective on the Main Machine Learning Methods Applied to Physical Sciences

Iris Iddaly Méndez-Gurrola, Abdiel Ramírez-Reyes

Universidad Autónoma de Ciudad Juárez, Av. del Charro no. 450 Nte. Col. Partido Romero, CP. 32310, Cd. Juárez, Chihuahua, México, iris.mendez@uacj.mx; abdiel.ramirez@uacj.mx

Alejandro Israel Barranco-Gutiérrez

Cátedras CONACyT - TecNM Celaya, Antonio García Cubas Pte #600 esq. Av. Tecnológico, CP. 38010, Celaya, Gto. México, israel.barranco@itcelaya.edu.mx

Abstract: Several types of numerical simulations have been used over the years in the Physical Sciences, to advance the real-life problems understanding. Among the statistical tools used for this are, for example: Monte Carlo simulations, such mechanisms have been used in various areas, however, today another tool is used, Machine Learning, which is a branch of Artificial Intelligence (AI). This article reviews sets of work that encompass various areas of the Physical Sciences, to mention some such as particle physics, quantum mechanics, condensed matter, among many others that have used some Machine Learning mechanisms to solve part of the problems raised in their research. In turn, a Machine Learning methods classification was carried out and it was identified which are the most used in Physical Sciences, something that is currently done in very few studies, as it requires extensive review work. The analysis carried out also allowed us to glimpse which areas of the Physical Sciences use Machine Learning the most and identify in which types of journals it is published more on the subject. The results obtained, show that there is currently a good number of works that interrelate Machine Learning and the Physical Sciences, and that this interrelation is increasing.

Keywords: Machine Learning; Physical Sciences; review; interdiciplinary

1 Introduction

Machine Learning (ML) is a methodology aims to implement capable computational algorithms of emulating human intelligence by incorporating ideas of probability and statistics, control theory, information theory, neuroscience, among other. This has allowed successful applications in various fields, such as

artificial vision, robotics, entertainment, biology, medicine, among others, so physical science could not be the exception. ML is basically integrated by 3 major learning paradigms [1]:

- ***Supervised learning:*** It creates a model that relates the output variables with those of the input. This function is used later to make predictions. This paradigm is generally used for regression and classification problems.
- ***Unsupervised learning:*** It has the objective of obtaining groups, such that in each of them there are homogeneous instances, while the groups are heterogeneous among themselves. In this learning there is no information from the past, it is the model itself in charge of making its own divisions. The tasks that cover this type of learning are grouping, dimensional reduction, association.
- ***Reinforcement learning:*** The algorithm learns, not with the previous information that has been provided, but with its interaction with the world that surrounds it, therefore, feedback is produced that modifies and refines its behavior.

Figure 1 shows the 3 paradigms of comprehend ML and some of the most common methods used in each category.

In addition to these three categories, in the present work the methods described below are also considered:

- **Ensemble methods:** These use the idea of combining several predictive models (supervised ML) to obtain higher quality predictions than each one of the models could provide individually. The most popular ensemble algorithms are Random Forest, XGBoost.
- **Neural Networks and Deep Learning:** Neural Networks are a subset of techniques that are inspired by the operation of connectionist systems, they are therefore within ML, the objective of Neural Networks is to capture non-linear patterns in data by adding layers of parameters to the model. Now the term Deep Learning comes from a Neural Network with many hidden layers and encapsulates a wide variety of architectures, for best performance Deep Learning techniques require a lot of data and a lot of calculations.

ML methods are designed to exploit large data sets in order to reduce complexity and find new functions in the data. Various Machine Learning algorithms have been used in the Physical Sciences, there are studies in the condensed matter physics area, such as Carrasquilla and Melko [2] that use a neural network with condensed matter model labels of low temperature and high temperature phase. The training set was given by an equilibrium configuration of the model obtained from Monte Carlo simulations.

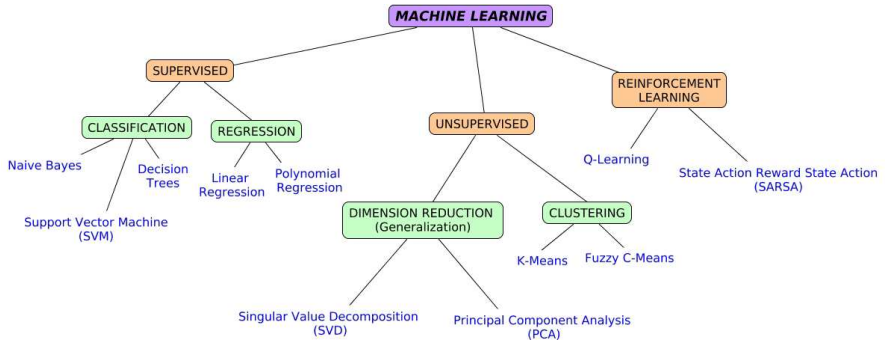

Figure 1
General ML scheme and some methods of each category

Another work used artificial neural networks to recognize different phases of matter and locate associated phase transitions, it is the work of Van Nieuwenburg, E. [3].

A review conducted in the area of particle physics by a research group [4] indicates that to date, the Large Hadron Collider (LHC) experiments have produced around 2,000 journal articles, providing a large library of examples for using ML with these kinds of complex data sets. In that work, for example, some highlights are discussed, including the role of ML in the discovery of the Higgs boson.

Challenges such as the 2014 Higgs Machine Learning Challenge and the Tracking Machine Learning challenge (TrackML) have even been carried out running on the Kaggle platform from March to June 2018 [5], which was a challenge for the ATLAS and CMS experiments, particularly for track reconstruction algorithms.

According to Zdeborová [6] every researcher in the Physical Sciences is clear that there are many numerical simulation types. Depending on the system and the interest question, knowledge and experience are required to find the correct numerical simulation and perform it carefully enough to be able to truly advance a given problem understanding. Under her consideration, the same goes for ML tool applications.

There are then various ML methods have been useful in the Physical Sciences, there are currently some review works such as Larkoski's [7] and Carleo [8] that show some ML methods applied to various areas, however, they do not perform a specific systematic review of a good number of methods used in various areas of the Physical Sciences. Therefore, the objective of this systematic review was to investigate the ML methods used in Physical Sciences, to detect the most used, in addition to identifying the Physical Sciences areas that the most take advantage of them, additionally to distinguish in what type of journals the most they are published.

The structure of this article continues with Section 2, where it explains the methodology used for the literature review. Section 3 shows the data analysis and research results. Finally, the last section addresses the conclusions and perspectives of the work.

2 Methods

2.1 Overview

For the representative review, the most relevant investigations were identified through a systematic search in various electronic resources, such as, ACM digital library, Annual Reviews, EBSCOHOST, IEEE Xplore digital library, Nature, ScienceDirect, Scopus, Wiley, Google Scholar, Web of Science and InSpire. These are 11 of the most used platforms for searching for information. The search ran through July 2021. The combined search terms included "Machine Learning", "Physical Sciences", "physics" and "review". The search was carried out with limited English language. Two of the reviewers performed the search independently, following the methodology of Snyder [9], titles, abstracts, as well as keywords were reviewed. The data collected from both searches was placed in a single directory.

In order to be included in this review, papers had to meet the following inclusion criteria: (1) be defined as research mentioning Machine Learning methods, (2) focus primarily on areas of Physical Sciences, (3) were included other previous review studies covering the intersection between ML and Physical Sciences, and (4) published in the period January 2005 and July 2021.

Referred scientific articles to any other area type not related to Physical Sciences, studies carried out entirely under a mathematical approach, as well as works that were complete books and those that were published in fields of knowledge other than physical and computational sciences they were excluded.

2.2 Data Extraction

The present article was carried out by two reviewers, the 11 electronic databases described above were used and 41 and 29 potentially relevant articles were found by each reviewer, giving a total of 70 articles in this phase. According to Snyder [9] the actual selection of the sample can be done in several ways, depending on the nature and scope of the specific review, in this case, the approach used was to perform the review in stages, firstly, the duplicate elements were deleted, leaving a total of 65 works, subsequently a preliminary analysis was carried out, based on the title, abstract and keywords, which reduced the number to 60, and finally the

exclusion criteria were applied, to select the documents that could be eligible for this study, a total of 55 articles remained in this selection (see Figure 2). Once we had selected elements, we proceeded with the collection of the articles in full text for detailed analysis.

For each article included the following variables were identified: (1) Machine Learning methods mentioned in the work, (2) areas of physics addressed, (3) if the work is a review or not, (4) the journal where it was published.

The classification was carried out by two of the reviewers, and although in general there was a great agreement in the information collected, there were some studies where the vision of a third reviewer was necessary to clarify some discrepancies and reach a consensus.

The search and selection process for relevant works is summarized in Figure 2.

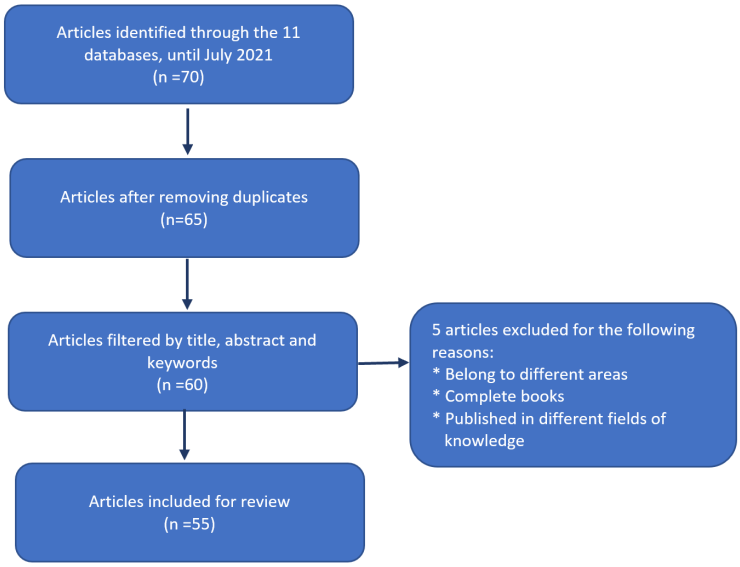

Figure 2
Search and selection strategy process flow chart

3 Results

3.1 Overview

Seventy records were identified through the electronic search carried out (see Figure 2). After removing duplicate items and applying the exclusion criteria, a total of fifty five articles were selected for further review. Once all the documents had been collected, it was necessary to identify the four variables described in the previous section for each work.

The following sections detail and explain the results obtained from the identification and classification of data according to the four variables, in particular Section 3.2 mentions the ML methods used or mentioned in each of the analyzed works, Section 3.3 mentions the areas of Physical Sciences that are covered in the articles, Section 3.4 describes the preceding review works and finally Section 3.5 illustrates in which type of journals they are most published, and a graph of the articles published as a function of time.

3.2 ML Methods

The articles analysis shows a great variety of ML methods, covering the 3 categories, both supervised, unsupervised and reinforcement learning, also including ensemble methods and deep learning. Table 1 shows each of the methods in detail, the total number of works where they were located and the main academic references where they are mentioned. It is important to note that in some studies more than one method is mentioned, even studies that included more than 10 methods were found, generally the review ones.

According to the results, it can be seen that the most used method is Neural Networks, which appears in 26 of the 55 references, which represents 47.27% of the total articles, in the second instance there are Convolutional Neural Networks that are mentioned in 21 articles, in third place we have both Vector Support Machines and Deep Neural Networks in 11 articles each, in fourth place are the Decision Trees that are mentioned in 10 articles and in fifth place are Generative Adversarial Networks mentioned in 8 works.

Among the less mentioned methods are Principal Component Analysis mentioned in 7 articles, followed by Random Forest mentioned in 6, and at the same level K-Nearest Neighbor and Recursive Neural Network with 5 mentions each, followed by three methods that have 4 mentions each, which are Gaussian Process Regression, K-Means, and Long Short-Term Memory.

Finally, it can be seen in Table 1 that there are several methods that are only named in one, two or three works, some of them are modifications or adaptations

to the main methods such as Physics-Guided Recurrent Neural Networks that combines RNN and models based in Physics to take advantage of their complementary strengths and improve physical process modeling [10]. Others of them are a new proposal such as Lift & Learn [11] which is a physics-based method to learn low-dimensional models for large-scale dynamical systems.

Table 1
Various ML methods found in analyzed works

Acronym	Meaning	Totals	Main Academic References
BDT	Boosted decision trees	3	[8] [12] [13]
BM	Boltzmann machine	2	[6] [8]
CNN	Convolutional neural network	21	[2] [4] [5] [6] [7] [8] [10] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27]
DBN	Deep belief network	1	[23]
DCN	Deep convolutional network	2	[8] [15]
DF	Decision forests	1	[6]
DNN	Deep neural network	11	[5] [7] [8] [12] [13] [19] [20] [23] [24] [28] [29]
DQL	Deep Q-learning	2	[16] [23]
DQN	Deep Q-networks	1	[23]
DRN	Deep residual network	1	[23]
DT	Decision trees	10	[16] [19] [20] [26] [30] [31] [32] [33] [34] [35]
EML	Ensemble Machine Learning	2	[32] [36]
GAN	Generative adversarial networks	8	[7] [8] [12] [15] [20] [23] [26] [29]
GBDT	Gradient-boosted decision trees	2	[35] [37]
GBRT	Gradient boosting regression trees	2	[8] [19]
GDL	Geometric deep learning	1	[5]
GPR	Gaussian process regression	4	[8] [25] [26] [38]
GRNN	Generalized regression neural network	1	[39]
GRU	Gated recurrent unit	1	[40]
K-means	K-means/medians	4	[16] [31] [41] [42]
KNN	K-nearest neighbour	5	[19] [20] [31] [33] [34]
LiR	Linear regression	1	[43]
LL	Lift & learn	1	[11]
LoR	Logistic regression	1	[19]

LSTM	Long short-term memory	4	[5] [10] [23] [40]
MCTS	Monte carlo tree search	1	[5]
MEM	Matrix element method	1	[12]
MLP	Multi-layered perceptron	1	[44]
MPR	Multivariate polynomial regression	2	[33] [45]
NB	Naive Bayes	2	[19] [20]
NN	Neural network	26	[3] [4] [6] [7] [8] [13] [14] [16] [20] [22] [26] [31] [32] [33] [34] [42] [43] [44] [46] [47] [48] [49] [50] [51] [52] [53]
PCA	Principal component analysis	7	[3] [8] [29] [34] [42] [47] [54]
PDML	Physics-driven Machine Learning	1	[55]
PGRNN	Physics-guided recurrent neural networks	1	[10]
PNN	Parsimonious neural networks	1	[56]
RBFN	Radial basis function network	1	[28]
ReF	Regression forests	1	[57]
RF	Random forest	6	[19] [26] [40] [41] [43] [53]
RM	Regression models	1	[16]
RNN	Recursive neural network	5	[7] [10] [15] [23] [40]
SVM	Support vector machine	11	[6] [8] [16] [18] [19] [23] [31] [32] [34] [37] [43]
SVR	Support vector regression	2	[40] [54]
VAE	Variational autoencoder	2	[7] [29]
XGB	XGBoost	1	[28]

3.3 Areas of Physics Addressed

In order to group the articles, the following classification was proposed, consisting of 2 categories that cover several areas of Physics. The first was Basic Physics and the second was Applied Physics. Figure 3 shows the classification made in this study and the percentage of works found by areas of Physics. It is important to note that all publications make use of ML as an extremely powerful tool to reach a conclusion or result in one or more areas of the Physical Scienes.

In Figure 3 it can be seen that the areas of knowledge concentrated in the publications are in first place Particle Physics (16%), followed by Materials Science (15%), in third place Quantum Mechanics (12%), later Condensed Matter (9%), Physical-Chemistry (6%), Atmospheric Physics (6%), Astrophysics (4%),

later, Mathematical Physics, Mechanics, Fluids, Statistical Physics, new Physics, Geophysics, Energy Systems with 3%, and finally with 1% each of the remaining areas.

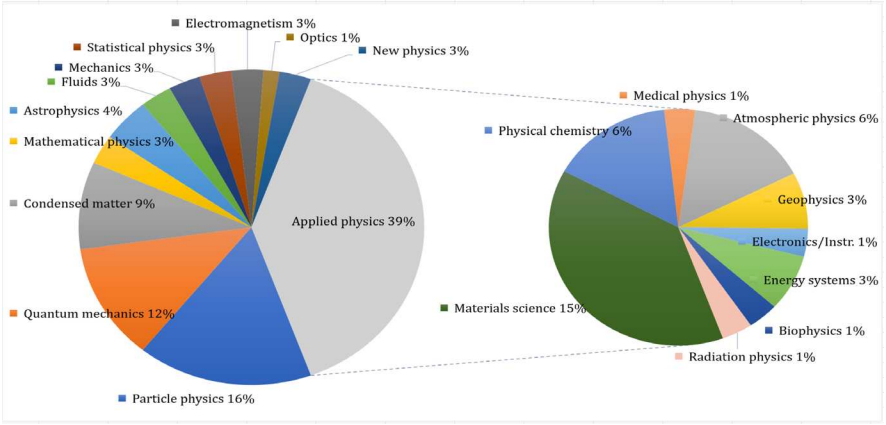

Figure 3
Areas of Physics that use Machine Learning

It is convenient to mention that it was possible to establish a methodology to identify the intersections of the Physical Sciences with Machine Learning, through which the publications in this regard were identified and based on this it becomes clear that Machine Learning is becoming an important tool in the Physical Sciences. This crossing is very novel, it can be explained by the ML strengthening and the use of GPUs. On the other one hand, due to the need for new tools to solve highly complex problems in the Physical Sciences, which alone cannot it is possible to solve them with traditional tools.

Something very interesting for the Physical Sciences is that, by qualitatively analyzing the works [30] [50] [51], a new strategy is observed to identify new Physics methods (such as new experiments in Quantum Mechanics or Physics beyond the model standard) using Artificial Intelligence. This is something completely new, Physics has never been built by AI. We can suggest the term AI Physics.

3.4 Review Articles

Most of the analyzed works are original articles, in which the result of an investigation is reflected with clarity and objectivity. On the other hand, 8 review works were located, some of them do not explicitly say “review”, however, they encompass a large number of works and show an overview of various aspects of ML methods applied in some fields of the Physical Sciences. These broadly contextualize the issue. Around 14.54% (n = 8) of the analyzed papers are review

articles, the methods mentioned in such papers are summarized in Table 2. As can be seen in the study by Carleo et. al [8], includes more than 10 methods, among them the most used as NN, GAN, CNN and SVM. It can also be seen that neural networks is the method most mentioned in the review papers.

Table 2
Review works and ML methods included in each of them

Study	ML methods mentioned
Larkoski et. al. [7]	NN, CNN, RNN,DNN,GAN, VAE
Carleo et. al. [8]	PCA, BM, GAN, NN, DNN, DCN, BDT, CNN, GPR, SVM, GBRT
Guest et. al. [15]	DCN, CNN, RNN, GAN
Radovic et. al. [4]	NN, CNN
Dunjko and Briegel [16]	NN, SVM, RM, K-means, DT, CNN, DQL
Zhang et. al. [32]	NN, SVM, DT, EML
Ng et. al. [43]	LiR, RF, SVM, NN
Cheng and Yu [23]	DNN, RNN, SVM, CNN, DQL, DQN, GAN, DRN, DBN, LSTM

3.5 Journals Where the Investigations were Published

The vast majority of the analyzed articles were published in Physical Science journals (see Figure 4), with 69% of the total, including: Physics Reports, Reviews of Modern Physics, Annual Review of Nuclear and Particle Science, Journal of Physics: Conference Series, Contemporary Physics, among many others. On the other hand, it can be observed in Figure 4 that only 20% of the investigations are in journals in the field of Computer Science, among them are: Applications of Artificial Intelligence, Procedia Computer Science, Computers & Fluids and some in as IEEE 14th International Conference on e-Science. However, 11% of the papers were published in journals considered interdisciplinary, such as: Nature communications, Scientific Reports, Nature.

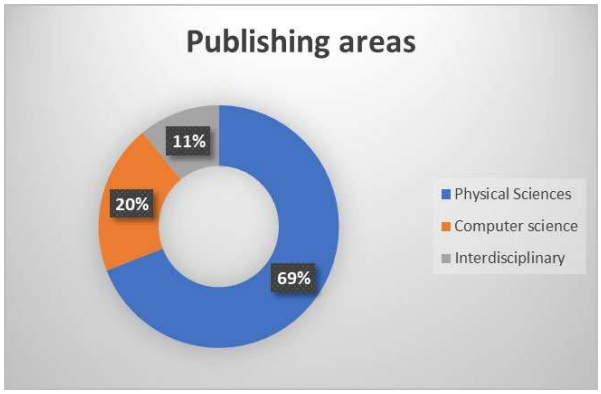

Figure 4
Journal types where it is mostly published

The publications are well referenced in databases that record articles of high academic quality; It is worth noting that several of the articles belong to the journals with the greatest impact in the Physical Sciences such as "Nature" [2-4] [6] [13] [20] [30] [35] [51]). Approximately 96% of these publications are cited in the “Journal Citation Reports 2020” [58].

3.5.1 Scientific Publications of ML and Physics as a Function of Time

Figure 5 shows the research articles analyzed in this work, as a function of time, covering the period January-2005 and July-2021. It highlights the significant and accelerated increase of publications in the last 5 years, which corresponds to 85% of the publications.

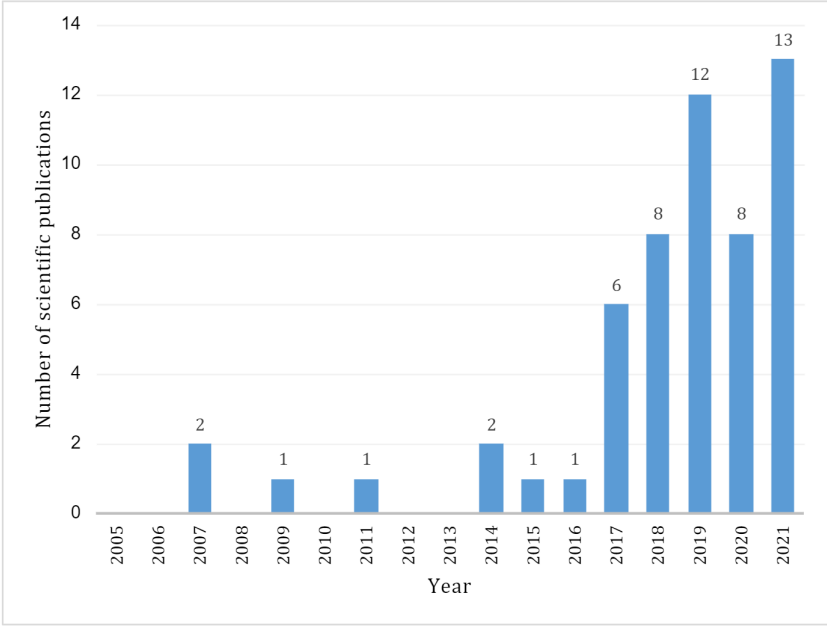

Figure 5
Number of publications in Physics using ML

Conclusions

In general, the results show that there is a good amount of work that connects Machine Learning and the Physical Sciences. It can be seen that there is a wide variety of ML methods that are used both for supervised and unsupervised learning, as well as for reinforcement. Those mentioned to a greater extent are Neural Networks, Convolutional Neural Networks, Vector Support Machines, Deep Neural Networks, Decision Trees and Generative Adversarial Networks. Some works that carry out new proposals were also found, such as Physics-Guided Recurrent Neural Networks that combines RNN and Physics-based

models and another such as Lift & Learn, which is a Physics-based method to learn low-dimensional models for large-scale dynamic systems.

On the other hand, according to the results, it can also be seen that there is an area of great variety of the Physical Sciences that use ML methods, among the most Particle Physics, Quantum Mechanics, Condensed Matter, which are considered within basic Physics. However, jobs were found within applied Physics, particularly in the areas of Materials Physics, Physico-Chemistry, among others.

Part of the research was to find other articles that were for review, few were found, in fact, just a total 8, which denotes that there is little research that considers the various areas of the Physical Sciences, most of the works are original articles, that are results of particular investigations.

Another contribution of this work was to differentiate the types of journals where the investigations were published, which helped us realize that most of them are from Physical Sciences and not, as could be expected, in the Computational area.

The work carried out shows that the interaction between Machine Learning and the Physical Sciences has shown growth in recent years, and this growth can be expected to continue in the coming years, especially in the areas of the Physical Sciences, where it has not yet been greatly applied, in order for interesting results to be generated.

References

[1] Nguyen, G., Dlugolinsky, S., Bobák, M. et al. Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev, 52, 77-124 (2019) https://doi.org/10.1007/s10462-018-09679-z

[2] Carrasquilla, J., Melko, R. Machine Learning phases of matter. Nature Phys 13, 431-434 (2017) https://doi.org/10.1038/nphys4035

[3] Van Nieuwenburg, E., Liu, Y. & Huber, S. Learning phase transitions by confusion. Nature Phys 13, 435-439 (2017) https://doi.org/10.1038/nphys4037

[4] Radovic, A., Williams, M., Rousseau, D. et al. Machine Learning at the energy and intensity frontiers of particle physics. Nature 560, 41-48 (2018) https://doi.org/10.1038/s41586-018-0361-2

[5] P. Calafiura et al., "TrackML: A High Energy Physics Particle Tracking Challenge," 2018 IEEE 14th International Conference on e-Science (e-Science), Amsterdam, 2018, pp. 344-344, doi: 10.1109/eScience.2018.00088

[6] Zdeborová, Lenka. Machine Learning: New tool in the box. Nature Physics, 2017, Vol. 13, No. 5, pp. 420-421

[7] Larkoski, Andrew J.; Moult, Ian; Nachman, Benjamin. Jet substructure at the Large Hadron Collider: a review of recent advances in theory and Machine Learning. Physics Reports, 2020, Vol. 841, pp. 1-63

[8] Carleo, Giuseppe, et al. Machine Learning and the Physical Sciences. Reviews of Modern Physics, 2019, Vol. 91, No. 4, p. 045002

[9] Snyder, Hannah. Literature review as a research methodology: An overview and guide-lines. Journal of Business Research, 2019, Vol. 104, pp. 333-339

[10] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S. Read, Jacob A. Zwart, Michael Steinbach, and Vipin Kumar. 2021. Physics-Guided Machine Learning for Scientific Discovery: An Application in Simulating Lake Temperature Profiles.Trans. Data Sci. 2, 3, Article 20 (May 2021)

[11] Qian, Elizabeth, et al. Lift & Learn: Physics-informed Machine Learning for large-scale nonlinear dynamical systems. Physica D: Nonlinear Phenomena, 2020, Vol. 406, p. 132401

[12] Albertsson, Kim, et al. Machine Learning in high energy physics community white paper. arXiv preprint arXiv:1807.02876, 2018

[13] Baldi, Pierre; Sadowski, Peter; Whiteson, Daniel. Searching for exotic particles in high-energy physics with deep learning. Nature communications, 2014, Vol. 5, p. 4308

[14] Sean Benson, Konstantin Gizdov, NNDrone: A toolkit for the mass application of Machine Learning in High Energy Physics, Computer Physics Communications, Volume 240, 2019, pp. 15-20, ISSN 0010-4655, https://doi.org/10.1016/j.cpc.2019.03.002

[15] Guest, Dan; Cranmer, Kyle; Whiteson, Daniel. Deep learning and its application to LHC physics. Annual Review of Nuclear and Particle Science, 2018, Vol. 68, pp. 161-181

[16] Dunjko, Vedran; Briegel, Hans J. Machine Learning & artificial intelligence in the quantum domain: a review of recent progress. Reports on Progress in Physics, 2018, Vol. 81, No. 7, p. 074001

[17] Ch'ng, Kelvin, et al. Machine Learning phases of strongly correlated fermions. Physical Review X, 2017, Vol. 7, No. 3, p. 031038

[18] Zhou, Jiajia, et al. Emerging role of Machine Learning in light-matter interaction. Light: Science & Applications, 2019, Vol. 8, No. 1, pp. 1-7

[19] Oviedo, F., Ren, Z., Sun, S. et al. Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks. npj Comput Mater 5, 60 (2019) https://doi.org/10.1038/s41524-019-0196-x

[20] Butler, Keith T., et al. Machine Learning for molecular and materials science. Nature, 2018, Vol. 559, No. 7715, pp. 547-555

[21] Li, A., Chen, R., Farimani, A. B. et al. Reaction diffusion system prediction based on convolutional neural network. Sci Rep 10, 3894 (2020) https://doi.org/10.1038/s41598-020-60853-2

[22] B. Y. Lattimer, J. L. Hodges, A. M. Lattimer, Using Machine Learning in physics-based simulation of fire, Fire Safety Journal, Volume 114, 2020, 102991, ISSN 0379-7112, https://doi.org/10.1016/j.firesaf.2020.102991

[23] Cheng L, Yu T. A new generation of AI: A review and perspective on Machine Learning technologies applied to smart energy and electric power systems. Int J Energy Res. 2019;43:1928-1973. https://doi.org/10.1002/er.4333

[24] Sigaki, H. Y. D., Lenzi, E. K., Zola, R. S., Perc, M., & Ribeiro, H. V. (2020) Learning physical properties of liquid crystals with deep convolutional neural networks. Scientific Reports, 10(1), 7664

[25] Watson-Parris, D. (2021) Machine Learning for weather and climate are worlds apart. Philosophical Transactions of the Royal Society A. Mathematical, Physical and Engineering Sciences, 379(2194)

[26] Hatfield, P. W., Gaffney, J. A., Anderson, G. J. et al. The data-driven future of high-energy-density physics. Nature 593, 351-361 (2021)

[27] Xu, D., Offner, S. S. R., Gutermuth, R., Oort, C. V. Application of Convolutional Neural Networks to Identify Protostellar Outflows in CO Emission (2021) Astrophysical Journal, 905 (2), art. no. 172. DOI: 10.3847/1538-4357/abc7bf

[28] Määttä, J., Bazaliy, V., Kimari, J., Djurabekova, F., Nordlund, K., & Roos, T. (2021). Gradient-based training and pruning of radial basis function networks with an application in materials physics. Neural Networks : The Official Journal of the International Neural Network Society, 133, 123-131

[29] Long-Gang Pang, Machine Learning for high energy heavy ion collisions, Nuclear Physics A, Volume 1005, 2021, 121972

[30] Krenn, Mario, et al. Automated search for new quantum experiments. Physical review letters, 2016, Vol. 116, No. 9, p. 090405

[31] Schuld, Maria; Sinayskiy, Ilya; Petruccione, Francesco. An introduction to quantum Machine Learning. Contemporary Physics, 2015, Vol. 56, No. 2, pp. 172-185

[32] Cheng Jun Zhang, Zhen Hong Shang, Wan Min Chen, Liu Xie, Xiang Hua Miao,A Review of Research on Pulsar Candidate Recognition Based on Machine Learning, Procedia Computer Science, Volume 166, 2020, pp. 534-538, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.02.050

[33] Swischuk, Renee, et al. Projection-based model reduction: Formulations for physics-based Machine Learning. Computers & Fluids, 2019, Vol. 179, pp. 704-717

[34] Ruta, D., Gabrys, B. A framework for Machine Learning based on dynamic physical fields. Nat Comput 8, 219-237 (2009) https://doi.org/10.1007/s11047-007-9064-6

[35] Hulbert, C., Rouet-Leduc, B., Johnson, P. A. et al. Similarity of fast and slow earthquakes illuminated by Machine Learning. Nature Geosci 12, 69-74 (2019) https://doi.org/10.1038/s41561-018-0272-8

[36] Mehnaz, Yang, L. H., Da, B., & Ding, Z. J. (2021) Ensemble Machine Learning methods: predicting electron stopping powers from a small experimental database. Physical Chemistry Chemical Physics : PCCP, 23(10), 6062-6074

[37] Rosenbrock, C. W., Homer, E. R., Csányi, G. et al. Discovering the building blocks of atomic systems using Machine Learning: application to grain boundaries. npj Comput Mater 3, 29 (2017) https://doi.org/10.1038/s41524-017-0027-x

[38] Maziar Raissi, George Em Karniadakis, Hidden physics models: Machine Learning of nonlinear partial differential equations, Journal of Computational Physics,Volume 357,2018, pp. 125-141, ISSN 0021-9991, https://doi.org/10.1016/j.jcp.2017.11.039

[39] Ling Qiao, Jingchuan Zhu, Yuan Wang, Coupling physics in Machine Learning to predict interlamellar spacing and mechanical properties of high carbon pearlitic steel, Materials Letters, Volume 293, 2021, 129645, ISSN 0167-577X, https://doi.org/10.1016/j.matlet.2021.129645

[40] Raden A. A. Ramadhan, Yosca R. J. Heatubun, Sek F. Tan, Hyun-Jin Lee, Comparison of physical and Machine Learning models for estimating solar irradiance and photovoltaic power, Renewable Energy, 2021

[41] Grazzini, Federico, et al. Extreme precipitation events over northern Italy. Part I: A systematic classification with machine-learning techniques. Quarterly Journal of the Royal Meteorological Society, 2020, Vol. 146, No. 726, pp. 69-85

[42] J. Xu, O. Schüssler, D. G. L. Rodriguez, F. Romahn and A. Doicu, "A Novel Ozone Profile Shape Retrieval Using Full-Physics Inverse Learning Machine (FP-ILM)," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 10, No. 12, pp. 5442-5457, Dec. 2017, doi: 10.1109/JSTARS.2017.2740168

[43] Ng, M., Zhao, J., Yan, Q. et al. Predicting the state of charge and health of batteries using data-driven Machine Learning. Nat Mach Intell 2, 161-170 (2020) https://doi.org/10.1038/s42256-020-0156-7

[44] S. Nakamura and S. Hashimoto, "Hybrid learning strategy to solve pendulum swing-up problem for real hardware," 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, 2007, pp. 1972-1977, doi: 10.1109/ROBIO.2007.4522469

[45] Arsenault, Louis-François, et al. Machine Learning for many-body physics: The case of the Anderson impurity model. Physical Review B, 2014, Vol. 90, No. 15, pp. 155136

[46] Shimon Whiteson, Daniel Whiteson, Machine Learning for event selection in high energy physics, Engineering Applications of Artificial Intelligence, Volume 22, Issue 8, 2009, pp. 1203-1217, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2009.05.004

[47] Rem, B. S., Käming, N., Tarnowski, M. et al. Identifying quantum phase transitions using artificial neural networks on experimental data. Nat. Phys. 15, 917-920 (2019) https://doi.org/10.1038/s41567-019-0554-0

[48] Zongrui Pei, Junqi Yin, Machine Learning as a contributor to physics: Understanding Mg alloys, Materials & Design, Volume 172, 2019, 107759, ISSN 0264-1275, https://doi.org/10.1016/j.matdes.2019.107759

[49] Matheus A. Cruz, Roney L. Thompson, Luiz E. B. Sampaio, Raphael D. A. Bacchi, The use of the Reynolds force vector in a physics informed Machine Learning approach for predictive turbulence modeling, Computers & Fluids, Volume 192, 2019, 104258, ISSN 0045-7930, https://doi.org/10.1016/j.compfluid.2019.104258

[50] D'agnolo, Raffaele Tito; Wulzer, Andrea. Learning new physics from a machine. Physical Review D, 2019, Vol. 99, No. 1, p. 015014

[51] Collins, Jack; Howe, Kiel; Nachman, Benjamin. Anomaly detection for resonant new physics with Machine Learning. Physical review letters, 2018, Vol. 121, No. 24, p. 241803

[52] Vandans, O., Yang, K., Wu, Z., & Dai, L. (2020). Identifying knot types of polymer conformations by Machine Learning. Physical Review. E,101(2-1)

[53] Pimachev, A. K., Neogi, S. First-principles prediction of electronic transport in fabricated semiconductor heterostructures via physics-aware Machine Learning. npj Comput Mater 7, 93 (2021) https://doi.org/10.1038/s41524-021-00562-0

[54] Zhu, Xiaofeng, et al. A planning quality evaluation tool for prostate adaptive IMRT based on Machine Learning. Medical physics, 2011, Vol. 38, No. 2, pp. 719-726

[55] Jici Wen, Qingrong Zou, Yujie Wei. Physics-driven Machine Learning model on temperature and time-dependent deformation in lithium metal and its finite element implementation, Journal of the Mechanics and Physics of Solids, Volume 153, 2021, 104481, ISSN 0022-5096, https://doi.org/10.1016/j.jmps.2021.104481

[56] Desai, S., Strachan, A. Parsimonious neural networks learn interpretable physical laws. Sci Rep 11, 12761 (2021)

[57] Kirkwood, C., Economou, T., Odbert, H., & Pugeault, N. (2021) A framework for probabilistic weather forecast post-processing across models and lead times using Machine Learning. Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences, 379(2194)

[58] Journal Citation Reports Social Sciences Edition (Clarivate Analytics, 2021) https://clarivate.com/webofsciencegroup/