OPEN ACCESS
Indo-JC Indonesia Journal of Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

IND. JOURNAL ON COMPUTING
VOL. 4, ISSUE. 3, DEC 2019. PP. 77-86
DOI:10.21108/INDOJC.2019.4.3.362

Entity Recognition for Quran English Version with Supervised Learning Approach

Muhammad Aris Maulana [#1], Moch Arif Bijaksana [#2], Arief Fatchul Huda [*3]

[#] *School of Computing, Telkom University*
Bandung, Indonesia

[1] muharismaulana@student.telkomuniversity.ac.id

[2] arifbijaksana@telkomuniversity.ac.id

[*] *Faculty of Science and Technology, State Islamic University (UIN)*
Bandung, Indonesia

[3] afhuda@uinsgd.ac.id

Abstract

The Quran is a Muslim holy book that consists of 6236 ayat or verses which divides into 144 surahs or chapters. In each chapter, there are many entities scattered in each verse. For a person, finding a particular entity will be difficult without a classification process, Resulting in difficulties in understanding the Quran. A system can be modeled to extract the information on entities in the Quran to solve this problem. Therefore, we want to offer a method to identify and classify entities using Entity recognition. The system will use the SVM techniques where the system will be given various entities from the Quran as an input to be able to identify correct entities. We are using the dataset obtained from website tanzil.net consists of 19.473 tokens and 720 entities. The classification scenario using a linear kernel with unigram produces the highest f-measure value of 0.75.

Keywords: Named-Entity Recognition, Quran, Supervised Learning.

Abstrak

Al-Quran merupakan kitab suci Muslim yang terdiri dari 6236 ayat atau bait yang dibagi menjadi 144 surah atau bab. Di setiap bab, ada banyak entitas yang tersebar di setiap ayat. Bagi seorang individu, menemukan entitas tertentu akan sulit tanpa proses klasifikasi yang membuat kesulitan dalam memahami Quran. Sebuah sistem dapat dimodelkan untuk mengekstrak informasi tentang entitas dalam Al-Quran untuk menyelesaikan masalah ini. Oleh karena itu, kami menawarkan sistem untuk mengidentifikasi dan mengklasifikasikan entitas menggunakan Entity Recognition. Sistem akan menggunakan teknik SVM di mana sistem akan diberikan berbagai entitas dari Quran sebagai input untuk dapat mengidentifikasi entitas yang benar. Kami menggunakan dataset yang diperoleh dari situs web tanzil.net terdiri dari 19.473 tokens dan 720 entitas. Skenario klasifikasi yang menggunakan linear kernel dengan unigram memperoleh nilai f-measure tertinggi sebesar 0,75.

Kata Kunci: Named-Entity Recognition, Quran, Supervised Learning.

I. INTRODUCTION

THE Quran, according to the *Kamus Besar Bahasa Indonesia* (Indonesian Dictionary) or KBBI is a Muslim holy book that contains the word of God revealed to the Prophet Muhammad. Through the angel Gabriel to be read, understood, and practiced as a guide or a way of life for humankind. The

RECEIVED ON NOV 2019. ACCEPTED ON DEC 2019

Quran is a holy book from the beginning until the time of its contents has not changed. This Muslim religious scripture consists of 6236 verses divided into 144 chapters. A chapter contains a different narrative between the other chapters and sometimes has no connection with one other chapter. In each of the chapters, there are also many entities scattered in each verse, which can make it difficult for someone to search for a particular entity. Entities in the Quran can refer to the name of people or groups. A classification system can be developed to make it easier for someone to find a particular entity.

Therefore, the researcher offers a technique for finding entities using Entity Recognition through the Supervised Learning approach to facilitate finding specific entities. Unlike the NER (Named-Entity Recognition) which classifies a named-entity, Entity Recognition limited to entities. The system overview can be seen in Fig. 1. The use of a dataset needed by following the Supervised Learning approach where the program will first be trained to achieve optimum accuracy where previously the dataset has been preprocessed. The primary input is the verses of the Saheeh International Quran version. The researcher chose The Saheeh International because it provided convenience to students, and brought readers closer to their original nuances [1]. After the *preprocessing* process, the entity that has been identified by the system will be classified into the entity class by taking the multiple-choice approach. To be more clearly given an example of an input of a verse in chapter As-Saff verse 6 "*when Jesus, the son of Mary, said," O children of Israel, indeed I am the messenger of God to you confirming what came before me of the Torah and bringing good tidings of a messenger to come after me, whose name is Ahmad. "But when he came up with clear evidences, they said," This is obvious magic. ""*, then detected "Jesus the son of Mary", "Ahmad" and "Children of Israel" which classify as entities. After that, the system builds a model that can classify entities in the Quran and make it easier for someone to understand the Quran. In conducting the research, researchers only use the Quran with Saheeh English translation taken through the website *tanzil.net*. The data obtained by us is still in the form of textit raw and unstructured. Therefore, preprocessing needs to be done first.

II. Literature Review

A. Named Entity Recognition

Named Entity Recognition or NER is a task in linguistic computing to find and classify the words of each word in a text or document that is not structured into a category such as people, organizations, locations, or monetary values. [2]. Categorizing NER into the process of Information Extraction. NER is a sophisticated system that works almost like the efficiency of the human brain. The system is structured in such a way that it can find entity elements from raw data and can determine the categories in which the elements are located. The system reads sentences and marks important entity elements in the text. The NER system has its uniqueness depending on the task or project provided. Therefore, it can hardly be used on different projects or systems.

In this research the researcher will identify the entity in the Quran solely based on an English translation version that uses Latin letters, not like in Arabic letters where the process of identifying entities is quite complex because Arabic letters do not use capital letters making it difficult to identify the name or group name. Thus, the translated version makes it more convenient.

According to Collins Dictionary of English, an entity is something that exists separately from other things and has a clear identity of its own. For this research, the researcher aims to identify an individual or a group that existed from the Quran perspective.

Several studies have been done related to this research, one of them is "Named entity recognition from biomedical text using SVM" which is one of the basic task s of biomedical text mining, of which purpose is to recognize the name of the specified type from the collection of biomedical text. NER result is usually the processing object of other text mining. NER from biological text is the foundation of bioinformatics research. In the experiment, the studies get precision rate= 84.24% and recall rate=80.76% [3].

Another study worth mentioning is the Arabic Named Entity Recognition: An SVM-Based Approach which identifying NER on Arabic verses and letters using ACE 2003, 2004 and 2005 corpora. By

Combining all the features, the system yields an F1=82.71 [4].

B. Supervised Learning

Supervised Learning is one part of machine learning to study functions that map an input to output based on examples of input-output pairs [5]. It can also be said as a function that works based on the data train that has been labeled [6]. The first step in Supervised Learning is to collect datasets [7] and choosing the correct features. An expert is needed to determine the best or the most informative field (attributes, features). If not, then the simplest way is to "brute-force," which measures the overall data in the hope that the right features can be found. The second step is data preparation and data pre-processing where the data is processed if it has missing value or outlier.

C. Support Vector Machines

Support Vector Machines is an algorithm that can be considered to be new in the Supervised Learning technique that was proposed by Vapnik, *et al* in 1995 [8]. The primary purpose of Support Vector Machines or SVM is to find the best *hyperplane* as a two classes separator with the equation in the formula 1 where w dan w_0 are parameters. The *Hyperplane* is used to maximize the margin between data that has the closest positive and negative class of *hyperplane* so that the data closest to hyperplane is also called *support vector*.

$$w^T \vec{x} + w_0 = 0 \tag{1}$$

This classification method is chosen based on a previous study [9] where it states that SVM produces a better performance against k-NN classifiers, Naive-Bayes Classifiers, and C4.5 Decision Tree Classifiers when applying classification with high dimensional data. The following is a basic form of SVM, as follows:

$$(w.x) + b = 0 \quad w \in \mathbf{R}^n, b \in \mathbf{R} \tag{2}$$

In SVM algorithms, also known a *kernel*, which is a set of mathematical functions. The function of *kernel* is to take data as the input and then transform it into the desired form [10]. *Kernel* also very suitable in high dimensional data, which why it is important for the research. There are several popular *kernels*, as follows:

1) Linear
   The linear kernel is used when the data is Linearly separable, which means a single line can separate the data. A large number of features of the data are suitable for linear use based on the cost of the computation, which is relatively low [11].

$$K(X_1, X_2) = X_1 \cdot X_2 \tag{3}$$

2) Radial Basis Function (RBF)
   This *kernel* is very suitable if the class labels and attributes is *nonlinear* [12]. RBF works by *nonlinearly* maps samples into a higher dimensional space. RBF uses normal curves around the data points and sums these so that the decision boundary can be defined by a type of topology condition such as curves where the sum is above a value of 0.5. The main drawback of RBF is the high costs of the computation process.

$$K(X_1, X_2) = \exp\left(-\gamma \parallel X_1 - X_2 \parallel^2\right) \tag{4}$$

3) Sigmoid
   Due to its origin from neural networks, the sigmoid kernel was quite famous for support vector machines. Where the bipolar sigmoid function is often used as an activation function for artificial neurons [13]. SVM model using a sigmoid kernel function is equivalent to a two-layer, perceptron neural network.

$$K(X_1, X_2) = \tanh\left(\gamma X_1 \cdot X_2 + c\right) \tag{5}$$

4) Polynomial
   Polynomial kernels are well suited for problems where all the training data is normalized. The Polynomial kernel is categorized as a *non-stationary* kernel.

$$K(X_1, X_2) = (ax_i^T x_j + r)^d \tag{6}$$

D. The Quran English Translated Saheeh International Version

Saheeh International is a translated version of the Quran conducted by three converts of American women published by Abul Qasim Publishing House, Saudi Arabia in 1997. The three main objectives of making the translation version are [1]:

1) To present correct meanings, as far as possible, in accordance with the *'aqeedah* of *Ahl as-Sunnah wal-Jama'ah*
2) To simplify and clarify the language for the benefit of all readers
3) To let the Quran speak for itself, adding footnotes only where deemed necessary for explanation of points not readily understood or when more than one meaning is acceptable

The researcher chose this version based on the popularity of the version. Also, it is the default translation version of English in tanzil.net and many other websites.

E. Evaluation System

Evaluation system is intended to determine how accurate the system is in classifying. The calculation is done by calculating the amount of data classified as true compared to all data. In the formula 9 there is an equation of *F-Measure* wherein the value of *precision* will be calculated using the formula 7 and *recall* that use the formula 8 [14].

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{7}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{8}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

III. RESEARCH METHOD

A. Dataset

The data used is a Quran verses obtained from *tanzil.net/trans/* in *text* file format. The obtained data is still not in the appropriate form in Table IV, therefore, it can not be directly used as a dataset. Some preprocessing needs to be done beforehand. Before applying the preprocessing, the obtained data then converted into a *csv (comma separated value)* format to make the preprocessing steps more convenient. After applying the preprocess, the researcher then manually labeled the token with a raw labeling technique inside the dataset. Furthermore, raw labeling is done, and the dataset is ready to be used. Fraction of dataset is provided in Table V. After that, the dataset is divided into two and distributed by using the rule of thumb, namely training as much as 80 percent from the whole dataset and testing as much as 20 percent.

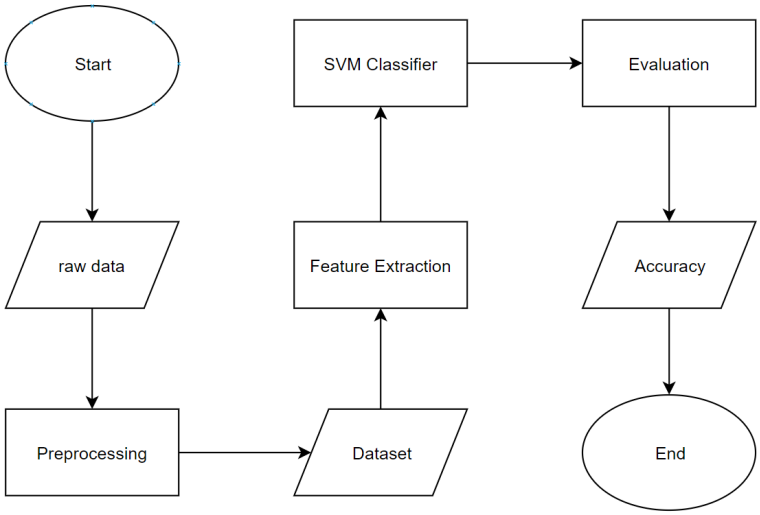

Fig. 1. The system overview of Entity Recognition in the Quran English Translation with Supervised Learning Approach.

TABLE I
RAW DATASET

Chapter	Verse	Surah	Text
...	...	...	...
1	1	Al-Fatihah	*In the name of Allah, the Entirely Merciful, the Especially Merciful.*
1	2	Al-Fatihah	*[All] praise is [due] to Allah, Lord of the worlds -*
1	3	Al-Fatihah	*The Entirely Merciful, the Especially Merciful,*
...	...	...	...

B. Preprocessing

Preprocessing techniques are chosen based on the form of data obtained from the source. Several preprocessing techniques that used to the system, as follows:

1) Punctuation Removal: In Table II, it can be seen that the process in this stage where a text containing symbols or punctuation will be removed, therefore making it convenient to build the system.

TABLE II
EXAMPLE OF PUNCTUATION REMOVAL

Input	Output
And [of] Abraham, who fulfilled [his obligations]	*And of Abraham who fulfilled his obligations*

2) Tokenization: In Table III, it can be seen that the process aims to break down each word in the document into a word that marked with the sign "," which is followed by blank spaces.

TABLE III
EXAMPLE OF TOKENIZATION

Input	Output
And of Abraham who fulfilled his obligations	*, And , of , Abraham , who , fulfilled , his , obligations ,*

3) Raw labelling: Raw labeling is a part of sequence labeling, and it works like Part of Speech or POS-tagging where each element gets a single tag. The standard way to do is using the "BIO" encoding (short for beginning, inside and outside) continued by "PER" which represents "Person". Each token then labeled by either "B-PER", "I-PER" or "O". Furthermore, "B-PER" means the beginning phrase of an entity, "I-PER" means the continued phrase of an entity and "O" means not in a phrase of the entity.

Raw labeling is done manually by the researchers based on an expert. An example is provided in Table IV.

TABLE IV
EXAMPLE OF RAW LABELLING

Word	Label
the	B-PER
angels	I-PER
prostate	O
before	O
Adam	B-PER
so	O
they	O

It can be seen that in the phrase there are two entities, the entity consists of a single word like "Adam" gets labeled as "B-PER" and not continued by I-PER also, "the angels" gets labeled as B-PER continued by I-PER because it is a unit.

C. Feature Extraction

Features have an important role in improving system performance, especially in unstructured documents. After preprocessing is done on the datasets, feature extraction will then be performed to simplify the classification process on the system. The selection of features in this research was inspired by previous studies which are a study about "Named Entity Recognition using Support Vector Machine: A Language-Independent Approach" which generally developed the NER system with the supervised machine learning approach in their study [15].

Before the feature extraction, the data begin with one dimension. After the extraction, it increases up to 42 dimensions depends on the feature selected. Furthermore, the feature contains the *feature language independent* which includes Morphological-based and Context-based, as follows:

1) Morphological-based:

- Prefix Is a group of letters placed before the root of a word. For example, the word "unhappy" consists of the prefix "un-" [which means "not"] combined with the root (or stem) word "happy"; the word "unhappy" means "not happy."
- Suffix is a group of letters placed after the root of a word. For example, the word flavorless consists of the root word "flavor" combined with the suffix "-less" [which means "without"]; the word "flavorless" means "having no flavor."
- Unigram is pairing a couple of tokens into one with one word on the right and one word on the left respectively.

2) Context-based:

- Titlecase is a feature that detects a word that begins with a capital letter.

These features will be carried out into 4 test scenarios which shown in Table VIII. Also, the dependent language feature that will be used in this study is Part of Speech-Tag or POS-Tag.

D. SVM Classifier

This research uses a module from *sklearn*, namely *LinerSVC* and *SVC*, which is a tool for training and testing models rather than NER. The selection of this classifier module is based on convenient and complete use. There are 4 kernels included inside the module, which are *linear*, RBF, *sigmoid* and *polynomial*kernel. Researcher will compare all of the kernel with a multiclass strategy using *OVR* or *One-vs-Rest* on the *LinearSVC*. The classification process then will be carried out for each normalized token, using *DictVectorizer* which will utilize the features that have been previously defined as attributes, and convert them into vector form so that that classification can be done with *LinearSVC* and *SVC*. The breakdown process of each kernel will be presented, as follows:

TABLE V
FRACTION OF DATASET AFTER PREPROCESSING

Chapter	Verse	Token	POS-Tag	Label
...	...	...	...	...
2	34	*the*	DT	B-PER
2	34	*disbelievers*	NNS	I-PER
2	35	*And*	CC	O
2	35	*We*	PRP	O
2	35	*said*	VBD	O
2	35	*Adam*	NNP	B-PER
2	35	*dwell*	NN	O
2	35	*you*	PRP	O
...	...	...	...	...

1) LinearSVC
It is an improvement from the classic SVC with parameter *kernel= "linear"*, but implemented in terms of *liblinear* rather than *libsvm*. *LinearSVC* also scale better to large numbers of sample based on the improvement flexibility in the choice of penalties and loss functions. Several parameters used in this module, *multi-class* set to "ovr" based on the *one-vs-rest classifiers* usage, *dual* set to True because *n-samples* $<$ *n-features* and the the rest of parameters are set to *default*

TABLE VI
LINEARSVC PARAMETERS

Parameters	Value	About
C	1.0	penalty parameter of the error term
multi-class	"ovr"	Determines the multi-class strategy if y contains more than two classes
penalty	12	Specifies the norm used in the penalization.
loss	squared-hinge	Specifies the loss function, âĂŸhingeâĂŹ is the standard SVM loss
tol	1e-4	Tolerance for stopping criteria.
fit-intercept	True	Whether to calculate the intercept for this model.
intercept-scaling	1	a 'synthetic' feature with constant value equals to intercept-scaling
verbose	0	this setting takes advantage of a per-process runtime setting in a multithreaded context.
random-state	None	Random number generator to use when shuffling the dual coordinate descent
max-iter	1000	The maximum number of iterations to be run.

2) SVC
The implementation is based on *libsvm*. The fit time scales at least quadratically with the number of samples and could be impractical beyond tens of thousands of samples. Several parameters used in this module, *gamma* set to "scale" to pass the 1 / (n-features * X.var()) and the rest is *default*.

TABLE VII
SVC PARAMETERS

Parameters	Value	About
C	1.0	penalty parameter of the error term
kernel	"kernel"	Specifies poly" or "rbf" or"sigmoid to "kernel"
degree	3	Degree of the polynomial kernel function ("poly"). Ignored by all other kernels.
gamma	"scale"	Kernel coefficient for "rbf", "poly" and "sigmoid". Function, "hinge" is the standard SVM loss
coef0	0	Independent term in kernel function.
shrinking	True	Whether to use the shrinking heuristic.
probability	False	Whether to enable probability estimates.
tol	1e-3	Tolerance for stopping criterion.
verbose	0	this setting takes advantage of a per-process runtime setting in a multi-threaded context.
max-iter	1000	The maximum number of iterations to be run.
random-state	None	Random number generator to use when shuffling the dual coordinate descent

E. Evaluation

The evaluation process is carried out in 4 scenarios based on the *kernels* and each of the scenarios will run on several times according to the features in Table VIII. After the system process predicts the

entity, based on the label, it will calculate the performance and average of each label. The calculation for system performance uses a measurement of Precision, Recall, and F-measure. After the evaluation process is carried, then we can determine the best model for Entity Recognition.

TABLE VIII
FEATURE COMPARISONS

No.	Features
1	Prefix + Suffix
2	POS-Tag
3	Unigram
4	POS-Tag + Unigram
5	Titlecase
6	Prefix + Suffix + POS-Tag + Unigram
7	Prefix + Suffix + Titlecase
8	POS-Tag + Unigram + Titlecase
9	Prefix + Suffix + POS-Tag + Unigram + Titlecase

IV. RESULTS AND DISCUSSION

TABLE IX
TEST RESULTS WITH ALL THE SCENARIOS AND FEATURE COMPARISONS

Kernel	Features	Precision	Recall	F-Measure
Linear	Prefix + Suffix	0.74	0.44	0.55
	POS-Tag	0.76	0.58	0.66
	Unigram	**0.78**	**0.72**	**0.75**
	POS-Tag + Unigram	0.76	0.72	0.74
	Titlecase	0.74	0.44	0.55
	Prefix + Suffix + POS-Tag + Unigram	0.75	0.73	0.74
	Prefix + Suffix + Titlecase	0.75	0.45	0.56
	POS-Tag + Unigram + Titlecase	0.77	0.73	0.75
	Prefix + Suffix + POS-Tag + Unigram + Titlecase	0.75	0.73	0.74
RBF	Prefix + Suffix	0.75	0.41	0.53
	POS-Tag	0.78	0.60	0.67
	Unigram	0.86	0.58	0.69
	POS-Tag + Unigram	0.89	0.60	0.72
	Titlecase	0.76	0.41	0.53
	Prefix + Suffix + POS-Tag + Unigram	0.89	0.65	0.75
	Prefix + Suffix + Titlecase	0.76	0.40	0.52
	POS-Tag + Unigram + Titlecase	0.88	0.59	0.71
	Prefix + Suffix + POS-Tag + Unigram + Titlecase	0.89	0.65	0.75
Sigmoid	Prefix + Suffix	0.72	0.41	0.53
	POS-Tag	0.50	0.31	0.48
	Unigram	0.75	0.49	0.59
	POS-Tag + Unigram	0.76	0.53	0.63
	Titlecase	0.72	0.39	0.49
	Prefix + Suffix + POS-Tag + Unigram	0.75	0.53	0.63
	Prefix + Suffix + Titlecase	0.66	0.39	0.49
	POS-Tag + Unigram + Titlecase	0.75	0.53	0.62
	Prefix + Suffix + POS-Tag + Unigram + Titlecase	0.73	0.54	0.62
Polynomial	Prefix + Suffix	0.76	0.44	0.54
	POS-Tag	0.79	0.61	0.69
	Unigram	0.85	0.59	0.70
	POS-Tag + Unigram	0.88	0.60	0.72
	Titlecase	0.76	0.43	0.55
	Prefix + Suffix + POS-Tag + Unigram	0.90	0.64	0.74
	Prefix + Suffix + Titlecase	0.78	0.42	0.54
	POS-Tag + Unigram + Titlecase	0.87	0.61	0.72
	Prefix + Suffix + POS-Tag + Unigram + Titlecase	0.90	0.65	0.75

The results of the research are summarized in Table IX, the predictive and actual values of B-PER and I-PER are then averaged and then the value of recall performance, presicion and f-measure are

obtained. The evaluation in this research was carried out in 4 scenarios with single and multiple features in each scenario. The scenario is tested using the prefix feature, suffix, titlecase, unigram, POS-Tag and the combination of each feature. After all the scenarios are done, a comparison will be made between the kernels and its features.

Based on the test scenario in Table IX, it can be seen that the difference in value among scenarios is nearly identical and leaving only a tight gap. The linear kernel has better performance in all the feature testing and *sigmoid* has the worse performance, nearly all the feature testing has the lowest value. POS-tag also did not perform well due to the Quran structure and sentences which is different in the modern culture. The best value among the scenario is 0.75, but the most efficient with only one feature is Linear kernel using *unigram*. The excellent performance signifies that the data is *linearly separable*, it means that the *kernel* could classify the data with only one straight line. *Unigram* feature is suitable for the Quran based on the structure of the entity, most of the entity either begin with "those who are" or "the one who", this repeated pattern makes *unigram* is the best choice. RBF also worth consideration, the precision value is the best among all scenarios. This is because of the way that the kernel works that changes with distance from a location but took longer and took more cost on computational.

V. CONCLUSION

There are several features used such as prefixes and suffixes, titlecase, unigram and POS-tag. We already know that the dataset is *linearly separable* and now we can determine the best model to the Entity Recognition is by using the Linear kernel and *unigram* feature based on F1 score with 0.75. It is also proved that using the Support Vector Machines is a pretty good performance on identifying entities in the Quran.

Therefore, it is expected that further research can determine the best features that can improve the performance of the model, as well as the improvement on the dataset and the kernel.

REFERENCES

[1] Saheeh International (1997) "The Qur'an: English Meanings", Abul-Qasim Publishing House
[2] Zitouni, Imed. (2014) "Natural language processing of semitic languages." Berlin: Springer.
[3] Ju, Zhenfei and Wang, Jian and Zhu, Fei. (2011) Named entity recognition from biomedical text using SVM. In 2011 5th international conference on bioinformatics and biomedical engineering (pp. 1-4). IEEE.
[4] Benajiba, Yassine and Diab, Mona and Rosso, Paolo and others. (2008) "Arabic named entity recognition: An svm-based approach." In Proceedings of 2008 Arab International Conference on Information Technology (ACIT) (pp. 16-18). Amman, Jordan: Association of Arab Universities.
[5] Russell, Stuart J and Norvig, Peter. (2004) "Artificial intelligence: a modern approach. Malaysia." *Pearson Education Limited. Rycroft-Malone, J.(2004). The PARIHS framework-A framework for guiding the implementation of evidenceÇę based practice. Journal of nursing care quality.* **19** (4): 297–304.
[6] Mohri, Mehryar and Rostamizadeh, Afshin and Talwalkar, Ameet. (2018) "Foundations of machine learning." MIT press.
[7] Kotsiantis, Sotiris B and Zaharakis, I and Pintelas, P. (2007) "Supervised machine learning: A review of classification techniques." *Emerging artificial intelligence applications in computer engineering.* **10** (160): 3–24.
[8] Vapnik, Vladimir. (2013) "The nature of statistical learning theory." Springer science & business media.
[9] Joachims T. (1998) Text categorization with support vector machines: Learning with many relevant features. InEuropean conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.
[10] Theodoridis, Sergios (2008). Pattern Recognition. Elsevier B.V. p. 203
[11] Maji S, Berg AC, Malik J.(2008) Classification using intersection kernel support vector machines is efficient. In2008 IEEE conference on computer vision and pattern recognition (pp. 1-8). IEEE.
[12] Hsu CW, Chang CC, Lin CJ. (2003) A practical guide to support vector classification.
[13] Lin HT, Lin CJ. (2003) A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. submitted to Neural Computation.3:1-32.
[14] Powers, David Martin (2011) "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." Bioinfo Publications.
[15] Ekbal, Asif and Bandyopadhyay, Sivaji (2010) "Named entity recognition using support vector machine: A language independent approach." International Journal of Electrical, Computer, and Systems Engineering. 4 (2):155-170.