International Journal on
Advanced Science
Engineering
Information Technology

Vol.12 (2022) No. 2
ISSN: 2088-5334

Information Extraction of Compound-Protein Interaction from Scientific Paper using Machine Learning

Aulia Afriza [a], Muhammad Rheza Muztahid [a], Annisa [a], Wisnu Ananta Kusuma [a,b,*]

[a] Department of Computer Sciences, IPB University, Bogor, Indonesia
[b] Tropical Biopharmaca Reseacrh Center, Bogor, Indonesia
Corresponding author: [*] ananta@apps.ipb.ac.id

***Abstract*— Drug Target Interaction (DTI) is an important process in drug discovery that aims to identify useful compounds in treatment. DTI research is mostly found in databases and literature or papers. To obtain DTI information, another method such as information extraction is required to retrieve information related to DTI interactions. The information in the abstract of the research paper contains many compound sentences. This study performs regular expressions to identify compound sentences, text mining for information extraction, and classification using Bernoulli Naive Bayes. The research uses a collection of abstract documents, where 3.000 abstract documents will be arranged into 29.363 sentences. Sentences that the regular expression has parsed are matched using pattern matching and conducted by text pre-processing. Sentences resulting from text pre-processing stages are used as training datasets. We use 10- fold cross-validation to evaluate the model. This research obtained the best average accuracy value of 0.72 for using naive Bayes without regular expression for compound sentences and 0.76 accuracies for naive Bayes with a regular expression for single sentences. Furthermore, by applying the feature selection process for compound sentence data, we obtained an accuracy of 0.731 for the model without regular expressions and an accuracy of 0.7644 for the model with feature selection using regular expressions.**

***Keywords*—Drug target interaction; information extraction; text mining.**

Manuscript received 20 Nov. 2020; revised 24 Sep. 2021; accepted 29 Oct. 2021. Date of publication 30 Apr. 2022.

I. Introduction

Drug discovery is a chemical process (it can be a simple chemical process or a complex protein process) or a combination of several chemical processes to help reduce symptoms of the disease without causing side effects to patients [1]. Drug Target Interactions (DTI) have an important role in drug discovery [2]. Identifying new drugs and their targets or looking for protein compound interactions is difficult because of the relatively limited knowledge of the complex relationship between the chemical and genomic spaces [3].

Many organizations such as Russelllab, BioGRID, and PhIN have been collected DTIs (compound-protein interactions) data based on research that has been conducted and then stored it into the database. However, the rapid development of pharmacology makes knowledge about the compound-protein interaction more contained in the literature or research papers than stored in the database. Many pharmacology researchers have conducted research but have not input their findings into the compound-protein interaction database. Another method for a researcher who wants to find information related to Diabetes mellitus is based on the literature from other researchers one by one, which causes the research process to be time-consuming. So, the growth of literature archives makes us attempt to implement the text mining method to quickly extract compound-protein interaction information from various research literature quickly and easily. Text mining is extracting interesting information and knowledge from unstructured text. Technically, text mining can be interpreted as using automated methods to exploit the vast amount of available knowledge in text documents [4].

Some studies have applied the text mining method in the field of bioinformatics. Lung *et al* [5] conducted text mining research to obtain the interaction of protein compounds by Logistic Regression, Linear Discriminant Analysis, and Naive Bayes in looking for interactions between compounds and proteins from abstract data in PubMed. Liu *et al.* [6] applied text mining, RNN, and CNN models to find information on drug interactions from a knowledge base (DrugBank) and sciences articles (Medline).

In this research, we used abstract documents downloaded from PubMed. We used ChemDataExtractor and LingPipe to recognize compound and protein entities within a sentence. We classified the interactions into two classes, deciding compound-protein interaction, "positive" and "negative". In the literature we used as corpus, we found that most researchers rarely used the five classes of compound-protein interactions as found in Biocreative VI. Instead, researchers often explain a compound-protein interaction in complex ways. However, it can be interpreted as a negative or positive interaction. Thus, it is necessary to do research by classifying interactions into two classes, referring to the characteristics of interactions that may appear and based on the context of the sentence in the literature.

The classification processes of this research attempted to use machine learning methods Naive Bayes and Regular Expression to help predict compound-protein interactions in compound sentences contained in the abstract document. This is because the abstract document we used in this research contained compound sentences. According to Yamamoto *et al.* [7], we can use regular expression techniques to identify compound sentences. A regular expression is a sequence of characters that describes a text pattern [8].

II. Material and Method

Information extraction aims to get information from natural language text [9] automatically. Various techniques have been proposed to extract from the relationship or interaction between two entities. The most common and direct approach is to use the classification method [10]. This research consists of several steps as follows:

- Sentences are parsing.
- Recognizing compound and protein entities in sentences
- Manual labeling, automatic classification processes, and validation
- Comparing the results of classification. The workflow of this research can be seen in Fig.1.

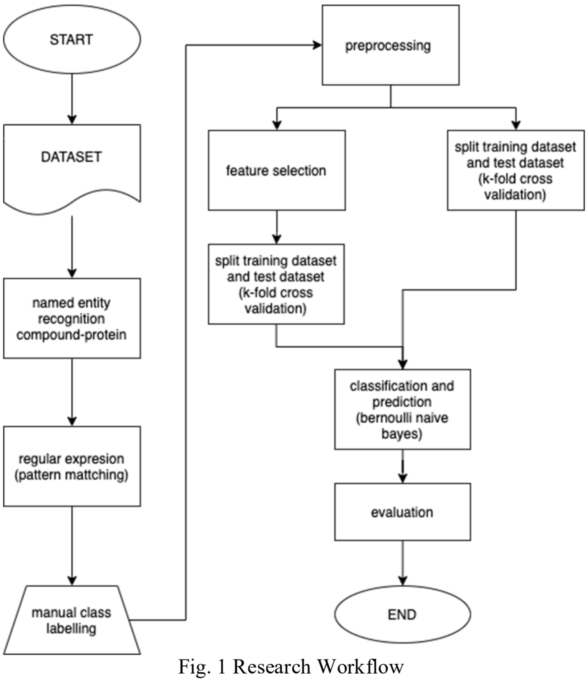
START
DATASET
named entity recognition compound-protein
regular expresion (pattern mattching)
manual class labelling
preprocessing
feature selection
split training dataset and test dataset (k-fold cross validation)
split training dataset and test dataset (k-fold cross validation)
classification and prediction (bernoulli naive bayes)
evaluation
END

Fig. 1 Research Workflow

A. Dataset

The dataset we used as the corpus is a collection of abstracts from research literature relating to Diabetes mellitus downloaded from PubMed (pubmed.gov). A total of 3,000 abstract datasets were downloaded from the PubMed database. This dataset was selected documents previously used by researchers in chemometric or bioinformatics as reading material for their studies.

B. Sentence parsing

Sentence parsing is used to separate each sentence in the dataset. The process of sentence parsing from 3,000 abstract datasets produced 29,363 sentences. These sentences are stored in a table for the next process to recognize each sentence's compound and protein entities. The examples of sentence parsing results can be seen in Table 1.

TABLE I
Example sentences parsing

Corpus	Sentences
14522431.txt	A simultaneous assay of glucose showed that the increase in insulin level was associated with a decrease in glucose level
7657022.txt	Improvements in GCR involve enhancement of insulin-mediated increase in muscle blood flow and the ability to extract glucose

C. Named Entity Recognition

Named Entity Recognition (NER) is used to assign the entity label of each word in the sentence [11]. In this study, NER was used to identify compound and protein entities. NER process was conducted using a python programming language. We used ChemDataExtractor and LingPipe for identifying the compound and gene/protein names.

ChemDataExtractor is a toolkit for automatically extracting chemical information from scientific documents. Machine-learning methods such as conditional random fields were used in combination with custom dictionaries and rule-based parsing grammars to extract valuable information from each sentence [12].

LingPipe is a toolkit for text processing using linguistic computing published by Alias-i, used to find people's names, organizations, and locations [13]. Carpenter succeeded in conducting research and building an additional function on LingPipe to be able to recognize gene/protein entities using the Hidden Markov Model (HMM) method [14].

The process was carried out with the 29,363 sentences, giving 7,653 sentences containing compound and protein entities. The remaining 21,710 sentences were not used in this research because they did not contain compound and protein entities. The collection of sentences containing compound and protein entities can be seen in Table 2.

TABLE II
NAMED ENTITY RECOGNITION RESULT

Corpus	Sentences	Chemical	Protein
10051433.txt	In addition to intracellular localization, bFGF is also widely distributed in the extracellular matrix, primarily bound to heparan sulfate proteoglycans (HSPGs).	sulfate	bFGF | HSPGs | HSPG
10051433.txt	To investigate this, we measured the effect of non-enzymic glycosylation on bFGF bound to heparin, heparan sulfate, and related compounds.	sulfate	bFGF

D. Regular expression

A regular expression is a key to powerful, flexible, and efficient text processing [15]. A regular expression is algebraic notation to characterize a series of strings [16]. Regular expression is useful for word-in-text search, where the regular expression search function will search through the data set and then return all text that matches the pattern [15]. The corpus can be a single document or a collection. The corpus can be a single document or a collection. The process of regular expression can be seen in Fig. 2.

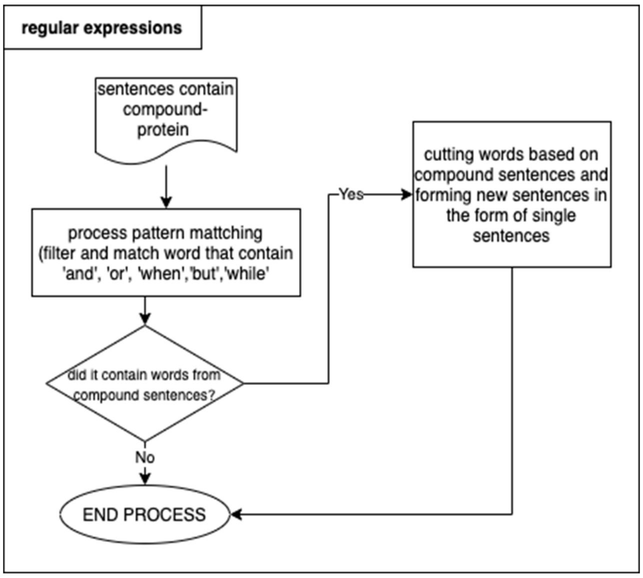
regular expressions
sentences contain compound-protein
process pattern matching (filter and match word that contain 'and', 'or', 'when','but','while'
did it contain words from compound sentences?
Yes
cutting words based on compound sentences and forming new sentences in the form of single sentences
No
END PROCESS

Fig. 2 Regular Expression Flow

E. Manual Labeling

The classification method is a type of supervised learning which can be defined as an approach in which trained data or labeled data exist to train the machine [17]. Manual labeling was done on 17,451 single sentences and 7,653 compound sentences. This process produced 3,869 negatives and 2,980 positives for single data sentences. For data contained compound sentences, this process produced 1,752 negatives and 1,057 positives. Then other sentences were not used to the next process.

F. Text pre-processing

Text pre-processing is a step of selecting the data to be processed in each sentence. This pre-processing includes case folding, tokenizing, filtering, and stemming [18]. After getting a set of sentences with compounds and proteins accompanied by the classification of interactions, we pre-processed the text on each sentence.

First, we conducted case-folding to change all the characters in the sentence to lowercase. Second, we separated each word with a "space" indicator to generate a set of words into an array called tokenizing. Tokenizing was carried out using Python by reading every word in each sentence. A single word is the smallest unit of a sentence separated by characters other than letters. Third, we filtered the information by removing unimportant words with a bag of stop words. Removing stop words was carried out in python programming using a stop list of 179 words available in the python programming language. Examples of words in the stop list are them, until, other, etc. The last step is stemming, which reduces inflected words to their word stem or basic form. Example words have added additives such as -ing, -s, -es, and others. The pre-processing text results from 6,849 single sentences produced a unique term of 5,669 words. Then, this process also succeeded in reducing the number of unique words from 2,809 words in compound sentences, and the remaining 5,542 unique words from data contained in compound sentences will become features in the next process.

G. Classification and validation

The data used in this section were generated from manual labeling and text pre-processing. Before conducting automatic classification and validation, the first step is to separate the training and the test data. The k-fold validation is a method for dividing the training and test data. In this study, we used the k-fold validation with k=10. It divided data randomly into k sections with the same amount of data. The process continues to cycle up to k times while changing the test section one by one until the tests are carried out in all parts [19].

Bernoulli Naive Bayes classification is a classification where the document is represented by a binary attribute that indicates the presence or absence of terms in the document [20]. In Kowsari *et al.* [20], the frequency of occurrence of terms in documents is not considered. When calculating the probability of a document, all attribute values are multiplied, including the likelihood and absence terms in the document. In this model, the word or term is assigned a value of 1 if it had a frequency of more than 0 in the Document Term Matrix or 0 if the word or term frequency is not included in the Document Term Matrix. The probability of d documents in class c is calculated using a formula [20]:

$$P(c\,|\,d)\alpha\ \ p(c)\prod\nolimits_{1\le i\le M} P(U_i = e_i\,|\,c) \qquad (1)$$

Where Ui is a random variable for vocabulary term i, note that the term is 0 (absent) or 1 (present) in class c. For example, the term will appear several times in class c documents, but only counts its occurrence in class c, does not count the number of terms that appear in class c, while P(c) is the probability of documents in class c. Estimates of P(c) and P (ei|c) are calculated using the equation [20]:

$$P(c)=\frac{Nc}{N},P(e_i\mid c)\frac{Nct}{Nc} \qquad (2)$$

Where Nc is the number of training documents in class c, while Nct is the number of documents that contain the term t in the class c training document. To eliminate the P (ei|c) estimate, which is zero in the equation, Laplace smoothing or Add-One Smoothing is used so that the estimation of P (ei|c) can be seen in the following equation [16]:

$$P(e_i\mid c)=\frac{Nct+1}{Nc+B} \qquad (2)$$

While B is the number of classes or categories [20], the evaluation stage is the stage to determine the level of accuracy and performance of the classification results. Classification performance was evaluated by calculating precision, recall, accuracy, and F-measure values. Precision is the level of accuracy between the information that is requested by the user and the answer given by the system. In comparison, recall is the system's success rate in rediscovering information [21]. Accuracy is defined as the level of closeness between predictive values and actual values [22].

F-Measure is one of the evaluation calculations in information retrieval that combines recall and precision. F-measure is commonly used as a standard balance between precision and recall evaluating the classification point. In fact, f-measure can be seen as an alternative point of AUC classification (Area Under the ROC curve) [23]. These values were compared to see the performance of the classification methods.

Receiver operating characteristics (ROC) is generally used to analyze the performance of classifiers in data mining [24]. The ROC graph is a 2-dimensional graph, with the false positive rate (FPR) plotted on the X-axis and the true positive rate (TPR) plotted on the Y-axis. A ROC graph describes the relative tradeoff between true positive and false negative [25].

An indication of the overall diagnostic accuracy of the ROC curve is the area under the curve. The area under the ROC curve (AUC) was used to summarize the entire ROC curve location. This is an effective measurement that combines the false positive rate (FPR) and the true positive rate (TPR) to describe the inherent validity of the experiments performed. The AUC value is 0 to 1; closer to 1 illustrates better results [26].

III. Result and Discussion

Bernoulli Naive Bayes classification uses the TF-IDF (Term Frequency - Inverse Document Frequency) value in calculating the probability value of the word frequency that appears from each dataset in each class. The classification was carried out in 4 steps below.

A. Bernoulli Naive Bayes without Regular Expression

First, we tried classification using Bernoulli Naive Bayes without regular expression to see the result. We used compound sentences for the dataset in this step. In this research, we used 10-fold cross-validation to divide the data into training and test data randomly, where we divided 90% or 1,880 of the dataset as training data and 10% or 209 of the dataset as test data. Bernoulli Naive Bayes achieved an average accuracy value of 0.72. The results for each model classification can be seen in Table 3.

TABLE III
Result accuracy, precision, recall, and F measure without regular expression

Model	Accuracy	Precision	Recall	F-Measure
K1	0.72	0.85	0.65	0.74
K2	0.67	0.83	0.55	0.66
K3	0.75	0.83	0.71	0.76
K4	0.75	0.78	0.80	0.79
K5	0.71	0.69	0.87	0.77
K6	0.72	0.71	0.86	0.78
K7	0.70	0.69	0.85	0.76
K8	0.70	0.7	0.81	0.75
K9	0.73	0.75	0.78	0.76
K10	0.77	0.78	0.83	0.80

The K10 model provided the best accuracy value of 77% with 78% precision, 83% recall, and 80% F-measure. The second step, we tried to add a step to separate compound sentences into one sentence.

B. Bernoulli Naive Bayes with Regular Expression

We used the regular expression process in this step, choosing pattern matching to separate compound sentences. The regular expression process was added before manual labeling, where the pattern matching process was performed with Python using the anaconda tools. The regular expression process is carried out to break down compound sentences with ambiguous classes, but if they are split into several sentences, they will become sentences with more specific classes.

At the parsing stage, 7,653 sentences were produced having compound sentences. After the regular expression process, sentences changed from 7,653 to 17,451 single sentences containing compound and protein entities. This process was carried out with a pattern search flow that represented compound sentences equivalent to 'and', 'or', 'but', 'while', and 'when'. After the pattern of the words was found, the sentence was cut from a matching pattern and sentence formation by looking for the subject and predicate to form a complete sentence. An example of cutting compound sentences into single sentences can be seen in Table 4.

TABLE IV
Example process regular expression

Compound Sentences	Subject	Predicate	Single Sentences
The effect of the extract was potentiated by 16.7 mM-glucose and 10 mM-L-alanine but not by 1 mM-3-isobutyl-1-methylxanthine.	The effect of extract	was potentiated	The effect of the extract was potentiated by 16.7 mM-glucose. The effect of extract was potentiated 10 mM-L-alanine. The effect of the extract was potentiated not by

The activity of the extract was found to be heat stable, acetone soluble, and unaltered by overnight exposure to acid (0.1 M-HCl) or dialysis to remove components with molecular mass < 2000 Da.	The activity of the extract	was found to be	1 mM-3-isobutyl-1-methylxanthine. The activity of the extract was found to be unaltered by overnight exposure to acid (0.1 M-HCl) The activity of the extract was found to be dialysis to remove components with molecular mass < 2000 Da.

After processing regular expression, the next step was manual labeling, text pre-processing, and classification. The data was divided the same as the first step classification, but for the test data, we used 685 and 6.164 for training data. Bernoulli Naive Bayes achieved an average accuracy value of 0.76. The results for each model classification can be seen in Table 5.

TABLE V
RESULT IN ACCURACY, PRECISION, RECALL, AND F MEASURE WITH REGULAR EXPRESSION

Model	Accuracy	Precision	Recall	F-Measure
K1	0.750	0.822	0.768	0.794
K2	0.800	0.823	0.816	0.819
K3	0.749	0.819	0.765	0.791
K4	0.780	0.816	0.778	0.796
K5	0.760	0.806	0.784	0.795
K6	0.750	0.788	0.786	0.787
K7	0.750	0.835	0.759	0.795
K8	0.770	0.830	0.780	0.800
K9	0.780	0.830	0.790	0.810
K10	0.770	0.820	0.780	0.800

The K2 model provided the best accuracy value of 80% with 82.3% precision, 81.6% recall, and 81.9% F-measure. The second step classification gave a value of 3% greater than the previous one. Next, we tried to add feature selection to see the best result for each step.

C. *Bernoulli Naive Bayes using Feature Selection Without Regular Expression*

We also tried to use feature selection to select the features that contribute the most to our prediction variable. As we know, having irrelevant features can reduce the accuracy of the models and make our model learn based on irrelevant ones. Therefore, we ordered the features from higher to lower based on their TF-IDF value. Then we ran the classification process ten times from 10% to 100% of the features to get a perfect set of features. Each of these experiments was performed using a 10-fold validation and resulted in the average value of the measurement results obtained. The result of the accuracy of each feature selection can be seen in Fig. 3.

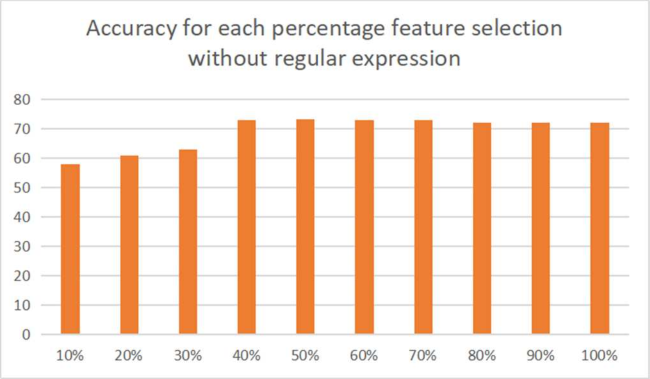

Fig. 3 Accuracy for each percentage feature selection without regular expression.

The best accuracy value was given at 73%, where this result was obtained from using 50% of the data of compound sentences. This step provided an accuracy value of 1% greater than the first one. Next, we tried to add feature selection using a regular expression to see the best result for each step.

D. *Bernoulli Naive Bayes Using Feature Selection with Regular Expression*

We also tried to use feature selection to see the accuracy when using a regular expression to get the comparison accuracy of each step. The accuracy of each feature selection with regular expression can be seen in Fig. 4.

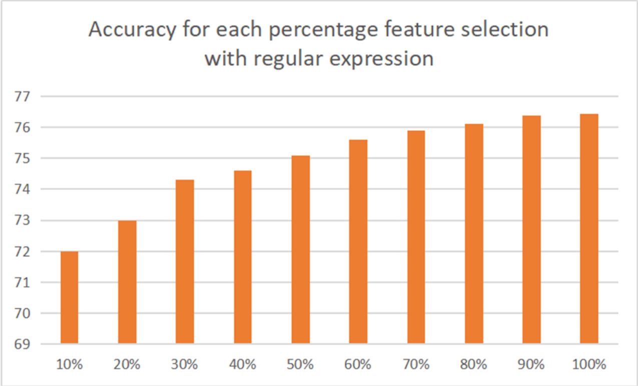

Fig. 4 Accuracy for each percentage feature selection with regular expressions.

The best accuracy value was given at 76.44%, where this result was obtained from using 100% of data of single sentences. This research provided the best accuracy value on the 100% feature selection model by using regular expression or using single sentences dataset. The increase in the accuracy value is based on adding regular expression with pattern matching, where in the first classification step, several compound sentences in a neutral class after separation compound sentences to a single sentence resulted in different classes as in the example in Table 4. The first example of Table 4 in the first step classification was in a neutral class, and after being separated into three single sentences, it produced two sentences belonging to the positive class and one sentence in the negative class. Those in the neutral class have compounds or proteins that do not have positive or negative interactions. If the TF-IDF value is ordered in 100% feature, the word order for the ten highest scores is shown in Fig. 5.

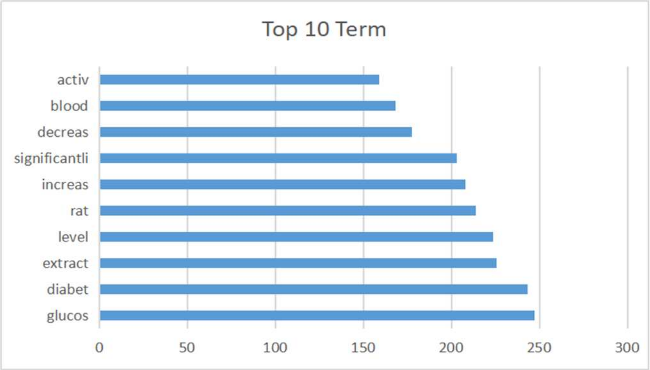

Fig. 5 TF-IDF value for the 10 terms with the highest value.

The results of the highest TF-IDF value can be seen that the words glucose, Diabetes, level, rat, and increase have the highest values. This word affects the classification results, where the classification Bernoulli Naive Bayes used the TF-IDF weight value in its calculation. The Bernoulli Naive Bayes classification calculated the TF-IDF weights of the combined distribution and then used the Bayes rule to calculate the posterior value.

The data characteristics have a unique term of 5,669 words using a single sentence for the dataset. The addition of unique terms occurred as many as 127 words from the dataset containing compound sentences, where the dataset containing compound sentences had 5.542 words. The additional step of a regular expression process with a matching pattern resulted in the addition step. In the pattern matching process, our example had one compound sentence, and we can be divided it into three single sentences, where the subject and predicate were taken based on the main compound sentence so that three single sentences had the same subject and predicate. This was because not too many unique terms were added.

The accuracy value was calculated as the ROC value [27]. The ROC curve was created by combining the false positive rate (FPR) and the true positive rate (TPR). The TPR value shows the proportion of positive interactions predicted by the system. Moreover, FPR shows how big the proportion of negative interactions is also predicted as negative interactions by the model. AUC or Area Under ROC Curve carried out the resulting TPR and FPR values. Area Under ROC Curve (AUC) is useful for summarizing all the locations of the ROC curve. The best accuracy value for the ROC curve before and after using regular expressions can be seen in Fig. 6.

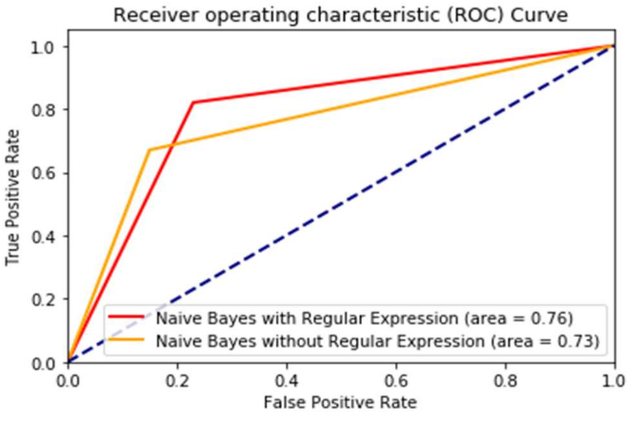

Fig. 6 ROC Curve

The accuracy value using a regular expression on the K10 model is 0.800 with an AUC value of 0.795, while the accuracy value of naive Bayes without regular expression is 0.731 with a value of 0.760. According to Nanda *et al.* [27], when seen in Fig. 2, the AUC value using regular expression is better than naive Bayes without regular expressions in terms of high sensitivity rate. A high sensitivity rate indicates a higher success rate in introducing a class.

IV. Conclusion

The Bernoulli Naive Bayes classification was carried out to extract the information about the compound-protein interaction in a collection of text. With 10-cross validation, this research obtained the best average accuracy value of 0.72 using Naive Bayes without regular expression for compound sentences and 0.76 accuracies using Naive Bayes with a regular expression for single sentences. Furthermore, by applying the feature selection process for compound sentence data, we obtained an accuracy of 0.731 for the model without regular expressions and an accuracy of 0.7644 for the model with feature selection using regular expressions. This study increased accuracy by 4% and 3.44% from Bernoulli Naive Bayes without using regular expressions and Bernoulli Naive Bayes using regular expressions, respectively.

References

[1] X. Xia. *Bioinformatics and Drug Discovery*. Bentham Sciences Publishers. 2017. pp. 1709-1726.

[2] L. Lee and H. Nam. "Identifications of Drug Target Interactions by a Random Walk with Restart Method on an Interaction Network". BMC Informatics Journal. vol. 19, no. 8, pp. 208. 2017.

[3] R. Chen, X. Liu, S. Jin , J. Lin , and J. Liu. "Machine Learning for Drug Target Interaction Prediction". Journal Molecules. vol. 23, pp. 2208. 2018.

[4] MU. Maheswari and JGR. Sathiaseelan. "Text Mining: A survey on text mining-techniques and application".International Journal of Science and Research (IJSR). vol. 6, pp. 1660–1664. 2017.

[5] PY. Lung, T. Zhao, Z. He, and J. Zhang. "Extracting chemical protein interactions from literature". Florida State University. 2018.

[6] S. Liu *et al.* "Attention Based Neural Network for Chemical Protein Relation Extraction". University at Buffalo USA. 2017.

[7] Y. Yamamoto, Y. Matsumoto, and T. Watanabe. "Dependency Patterns of Complex Sentences and Semantic Disambiguation for Abstract Meaning Representation Parsing". in *Proc, Conference on Lexical and Computational Semantics.* Bangkok, Thailand. 2021. pp. 212-221

[8] M. Cu et al. "Regular Expressions Based Medical Text Classification using Constructive Heuristic Approch". in IEEE Access, vol. 7, pp. 147892-147904, 2019, doi: 10.1109/ACCESS.2019.2946622.

[9] A. Konys. "Torwards Knowledge Handling in Ontology- Based Infromation Extraction Systems". in Procedia Computer Science. vol. 126, pp. 2208-2218. 2018.

[10] C. Zong, R, Xia, and J. Zang. *Information Extraction. In: Text Data Mining*. Springer, Singapore. 2021.

[11] J. Starvoka, M. Staraka, and J. Hajic. "Neural architectures for nested NER through Linearization". arXiv preprint arXiv:1908.06926. 2019.

[12] M, Erin *et al.* "Ensemble Labeling Towards Scientific Information Extraction (ELSIE)." ICCS. 2021.

[13] J, Savoy. "Working with Text Tools, Techniques, and Approches for Text Mining". Journal of the Assocation for Information Science and Technology. vol. 69. 2017

[14] B. Carpenter. "Lingpipe for 99.99% recall of gene mentions". Proceedings of the Second BioCreative Challenge Evaluation Workshop, BioCreative, pp. 307–309. 2007.

[15] Z. Zhong et al. "Generating Regular Expressions from Natural Language Specifications: Are We There Yet?". Association for the Advancement of Artificial Intelligence. 2018.

[16] L. G. Michael, J. Donohue, J. C. Davis, D. Lee and F. Servant, "Regexes are Hard: Decision-Making, Difficulties, and Risks in

Regular Programming Expressions," 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019, pp. 415-426, doi: 10.1109/ASE.2019.00047.

[17] C. Darujati and AB. Gumelar. "Pemanfaatan teknik supervised untuk klasifikasi teks bahasa indonesia". Jurnal Bandung Text Mining, vol. 16, no 1, ISSN. 1858–4667. 2012.

[18] YH. Kerner, D. Miller and Y. Yigal. "The influence of pre-processing on text classification using a bag-of-words representation." PLoS ONE. 2020.

[19] D. Abdelhafiz, C. Yang, R. Ammar, and S. Nabavi. "Deep convolutional neural networks for mammography: advances, challenges and applications". BMC Bioinformatics 20. 2019. https://doi.org/10.1186/s12859-019-2823-4

[20] K. Kowsari et al. "Text Classification Algorithms: A Survey.2019. https://doi.org/10.3390/info10040150.

[21] Heyong and H. Ming. "Supervised Hebb rule-based feature selection for text classification". in Information Processing & Management. vol. 56, pp. 167-191. 2019.

[22] DM. Powers. "Evaluation: from precision, recall and f-measure to roc, informedness, markedness, and correlation". International Journal of Machine Learning Technology. vol .2, pp.37-63. 2020.

[23] L. García-Bañuelos, N. R. T. P. van Beest, M. Dumas, M. L. Rosa and W. Mertens, "Complete and Interpretable Conformance Checking of Business Processes," in IEEE Transactions on Software Engineering, vol. 44, no. 3, pp. 262-290. 2018, doi: 10.1109/TSE.2017.2668418.

[24] Ghosh, M., Sanyal, G. An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning. J Big Data 5, 44. 2018. https://doi.org/10.1186/s40537-018-0152-5.

[25] Liang, L., Ma, C., Du, T. et al. Bioactivity-explorer: a web application for interactive visualization and exploration of bioactivity data. J Cheminform. 2019.

[26] A. Cecile, JW. Janssens, and FK. Martens, Reflection on modern methods: Revisiting the area under the ROC Curve, International Journal of Epidemiology, Volume 49, Issue 4. 2020, Pages 1397–1403.

[27] Nanda, M.A. *et al.* "A Comparison Study of Kernel Functions in the Support Vector Machine and Its Application for Termite Detection". 2018.