A Supervised Machine Learning Approach of Extracting Coexpression Relationship among Genes from Literature

Richa Tiwari, Chengcui Zhang, Thamar Solorio
Department of Computer and Information Sciences,
The University of Alabama at Birmingham, Birmingham, Alabama, USA
{rtiwari, zhang, solorio}@cis.uab.edu

Abstract

It is vital to develop automatic information extraction systems to help researchers cope up with the vast amount of data available on the Internet. In this paper, we describe a framework to extract precise information about coexpression relationship among genes, from published literature using a supervised machine learning approach. We use a graphical model, Dynamic Conditional Random Fields (DCRFs), for training our classifier. Our approach is based on semantic analysis of text to classify the predicates describing coexpression relationship rather than detecting the presence of keywords. We compared our results of sentence classification with the baseline technique of word matching and a Naïve Bayes classification algorithm. Our framework outperformed the baseline by almost 45%, with DCRFs showing superior performance to Naïve Bayes.

Keywords: Machine learning, Gene coexpression, Dynamic Conditional Random Fields, Relationship extraction.

1. Introduction

Information extraction applications have become a very important part of research in almost all areas nowadays. Massive volume of research data that are available on the Internet in various forms, such as text, are highly dispersed and manually extracting precise information from them is a very tedious task. Researchers in computer sciences are continuously developing methodologies that can help in managing and retrieving such huge corpora of publications and data. Information extraction techniques help in automatically retrieving precise, explicit as well as implicit information from text. The field of biomedical research is one of the areas where such techniques are seen to be very useful. For example, there is an overwhelming amount of information available in the form of text that can be mined for precise knowledge about certain gene or gene-disease relationships. Systems that automatically extract information such as relationship between proteins, different kinds of gene-gene interactions, and/or gene-disease relations have proven to be very useful in this area.

In this paper, we focus on one such type of relationship extraction, i.e., gene-gene coexpression relationship extraction. Two or more genes are co-expressed if they are expressed simultaneously into mRNAs and further translated to proteins. Coexpression relationships can help us identify other valuable information such as the pathways these genes are involved in and their functional properties. Usually, coexpression information can be extracted from DNA microarray data, which are publicly available in gene expression repositories [1]. However, these repositories only provide names of genes being coexpressed but do not give any other information such as the experimental results of the coexpression, e.g., whether there is any common regulatory element(s) between them or their cumulative effect. Such questions can be answered only from the papers that are published with the information regarding those coexpression experiments. To our knowledge, there is no existing tool that can extract predicates that talk about coexpression relationships between and among genes from the published articles based on text analysis and information extraction techniques. Our goal in this work is to automatically extract predicates from text that talk about coexpression relationships from published papers. This further helps in distinguishing papers that talk about coexpression of some particular genes from the papers that only talk about coexpression process or relationships in general. Once we are able to extract those relevant papers, it will be easier for the interested researchers to organize and retrieve answers for answering other coexpression relationships related questions from this smaller set of papers. Also, the extracted relationships can be further used to build up networks that can describe complex biological pathways.

We have used Dynamic Conditional Random Fields (DCRFs), a graphical probabilistic model, trained on these predicates, and later present our results by testing these models on full-length biomedical papers collected from the National Center for Biotechnology Information (NCBI) (www.ncbi.nlm.nih.gov).

978-1-4244-8099-9/10/$26.00 ©2010 IEEE

In Section 2, we discuss the related work in the field of information extraction and relationship extraction, and Section 3 illustrates our approach. We present our analysis of results in Section 4. Section 5 concludes this paper.

2. Related Work

Information extraction from text is a process of automatically extracting structured data from unstructured data, usually text written in a natural language like English. There has been a lot of work done in the area of information extraction in the field of biomedical sciences. This includes extraction of information such as named-entities, terminologies, relationships etc. In the past, researchers have used various natural language processing techniques such as hand-written rules based on linguistic knowledge, to extract information from text [2]. However, these approaches are very expensive and require a lot of time and effort from domain experts. Hence, more statistical approaches such as machine learning have been introduced to aid the Information Extraction task. Hidden Markov Models, Neural Networks and similar stochastic classification algorithms have been successfully used in extracting information from text [3, 4]. Probabilistic graphical models like Hidden Markov models (HMM) and Conditional Random Fields (CRFs) have proven to be quite successful in classifying and extracting relationships between biomedical entities. CRFs are undirected graphical models developed by Lafferty et al. in 2001, and have been applied to various text mining tasks such as table extraction and named entity extraction in biomedical text [5, 6]. Bundschus et al. used linear chain CRFs to extract gene-disease relationships from biomedical literature [7]. DCRFs are an extension of CRFs and have been shown to outperform CRFs in natural language chunking tasks [8].

Gene-gene coexpression relationship extraction is a subtask of relationship extraction, but differs in the way that this relationship is expressed in natural language in actual text. Generally there is a pattern of expressing a relationship between two entities, for instance the relationship word, usually a verb, will occur in between the entities in the text and hence can be extracted if we know the entity boundaries. While coexpression relationships are not necessarily written in this format, certain challenges can occur in classifying those using fixed patterns.

3. The proposed approach

Automatic relationship extraction from unstructured machine-readable text is a complex and challenging task. Following recent successes in using CRFs for relationship extraction, we have explored their use in our work. Relationship extraction is comprised of annotating the unstructured text with the entities involved and the relationship between those entities. In our case, entities are the genes and the relationship terms are the ones that express coexpression relationships. As illustrated in Figure 1, our entire framework can be divided into 5 steps: 1) Data collection, 2) Pre-Filtering, 3) Feature extraction and class assignment, 4) DCRFs Model Training, and 5) Feature and Class Analysis. In the subsections below we describe each of these steps in detail.

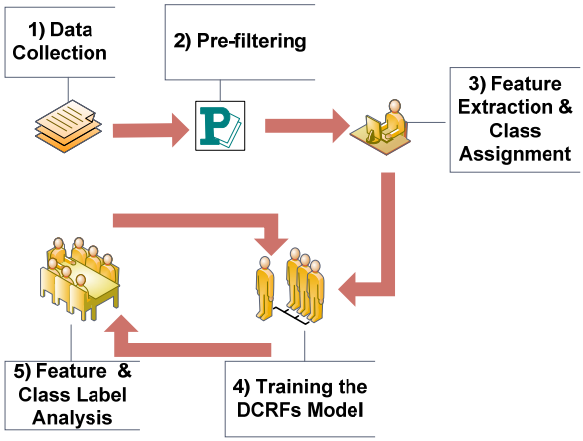

Figure 1. Graphical representation of our framework

3.1. Data collection

We collected 500 full-length papers from NCBI and manually divided them into positive and negative papers depending upon whether a paper contains predicates about some genes being coexpressed and not just the coexpression process in general. A paper that contains positive sentences has both gene names as well the terms defining the coexpression relationship in them. Figure 2 shows some examples of sentences that distinguish a positive paper from negative ones.

Positive Sentences

Functionally , coexpression of Kir2.1 and PSD-93 had no discernible effect upon channel kinetics but resulted in cell surface Kir2.1 clustering and suppression of channel internalization.

The coexpression of the c-Myb DBD dominant negative mutant protein was effective in blocking the up-regulation of all three promoters.

Negative Sentences

We further identified the partial co-expression relationship between genes: gene profiles may simultaneously rise and fall in a sub-range of the time course rather than the overall time course.

One important goal of analyzing gene expression data is to discover co-regulated genes.

Figure 2. Examples of positive and negative sentences.

3.2. Pre-filtering

The collected papers are all in PDF format and were converted into text using the PDFbox java library [9]. Once these papers were converted into text, we performed down sampling on them and deleted the sentences or parts from them that we believe would not provide any useful information about coexpression relationships. These parts include – everything before the abstract section of the paper and everything after the acknowledgment section of the paper. As mentioned earlier, relationship extraction includes a process of tagging the entities and the relationship among them and in our case these entities are gene names. Hence, we assume that the sentences that do not contain the desired entities (gene names) usually would not correspond to any useful information about the relationship. We consequently deleted all the sentences that do not contain gene names. To determine if any gene name is mentioned in a given sentence, we tagged each word in a sentence using the GENIA tagger, a freely available tool that has been trained using HMMs to identify gene names in any given text [10]. The sentences that are not tagged by the GENIA tagger as containing gene names are discarded from the succeeding steps.

3.3. Feature extraction and class assignment

Feature extraction is a very important task in machine learning techniques. It is important to provide a good set of features to any machine learning algorithm, if we want the model to perform well. However, there is no fixed set of features that can be used to improve the performance of any model, as they are very task specific.

The basic set of features is extracted using the GENIA tagger. GENIA tagger parses the sentences and tokenizes them into words. It then assigns features to each word. These features include, the root form of the word, e.g., the word *coexpression* has the root *coexpress*. It also performs grammatical analysis of each sentence and extracts the part of speech tag feature for each word. Part of speech (POS) feature of a word describes the grammatical role of the word in the sentence, e.g., the word "books" has a POS tag as *noun* in the sentence "Books are made of paper ink and glue" But the same word is tagged as *verb* for the sentence "book your flight soon". Another set of features that GENIA tagger extracts is often referred to as the *Chunk tags.* Chunk tags are the tags for the constituent phrases in a sentence. A constituent phrase is a group of words that functions as a single unit in a sentence. Therefore, all the words in a noun phrase will get the chunk tag as NP (Noun Phrase) as a feature. It is worth noting that a biomedical entity term will be assigned a chunk tag as 'B-NP' by the GENIA tagger. Finally as mentioned earlier, GENIA tagger also assigns biomedical entity tags to the entities, such as "B-protein" and "B-RNA". And those tags act as additional features of those words. The final basic feature set can be divided into two main categories:

Local:

- The word itself and the root of the word
- POS tag of the word and its chunk tag, i.e., the tag at the phrase level
- Biomedical named-entity tag

Contextual:

- POS and chunk tags of the words at positions -3,-2,-1, +1, +2, +3
- Biomedically named entity tags of the words at positions -3, -2, -1, +1, +2, +3

We train a DCRFs model on the class labels assigned to words in a way as follows:

- RE class labels assigned to the words that express the coexpression relationships
- GE class labels assigned to the gene name words that are involved in the coexpression
- Chunk tags assigned to the rest of the words in the sentence

Overall, we have 12 class labels including the GE, RE and all the chunk tags, out of which we are interested in classification of RE and GE labels by our model.

These are the basic or the original set of features and classes that we started our experiment with and later on we refined by analyzing the results in Steps 4 and 5.

3.4. Training the DCRFs model

We perform supervised learning using DCRFs, a probabilistic graphical model that can be effectively used for sequence classification problems. It is a generative model that relaxes the Markov assumption of HMM regarding the input and output sequence. DCRFs calculate the conditional probability distribution of output label sequences given a particular observation sequence $p(y|x)$ rather than finding the joint probabilities of both label and observation sequence [11]. Equation 1 shows how to calculate conditional probabilities of output labels given the input observations using DCRFs.

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{t} \prod_{c \in C} \exp\left\{ \sum_{k=1}^{K} \lambda_k f_k \left(y_{t,c}, x, t \right) \right\} \qquad (1)$$

where, f_k is referred to as the transition feature function, which is equivalent to transition probabilities in HMM. λ_k is the learned parameter vector in the model. $Z(x)$ represents the normalization function, and K is the number of feature functions. t is the time step (state) and C being the set of clique indices. A clique in an undirected graph G is a set of vertices V such that for every two vertices in V, there is an edge connecting them.

In our framework we have used GRMM a graphical model toolkit of mallet, which has a java implementation of DCRFs [12].

Once the input files are tagged with their true class labels and features, we divide the data into training and

testing sets. We also create a small subset of tagged data referred to as the development set, which consists of 4,109 sentences (1,333 positives ones and 2,776 negative ones) that is used for parameter tuning such as appropriate feature selection. This development set was used in the next step to select the appropriate set of features and class labels. In other words, the next step is to determine which features to use for predicting which class labels.

3.5. Feature and class selection

In this section we discuss our process of feature selection and class label analysis. We use our development set as the testing data for this process and performed several experiments with different feature combinations. DCRFs are good for representing complex interactions between class labels. In our framework, this will refer to the interaction between class labels for genes that are actually involved in coexpression relationships and the labels for coexpression relationship terms. This leads us to experiment with various sets of appropriate class labels because gene name extraction in itself is a big area of research and we are more interested in finding the coexpression terms. Finally we come up with a set of features and class labels that have the strongest correlation with the classification variable, i.e., the feature and class label combination that gives the best results for tagging RE labels.

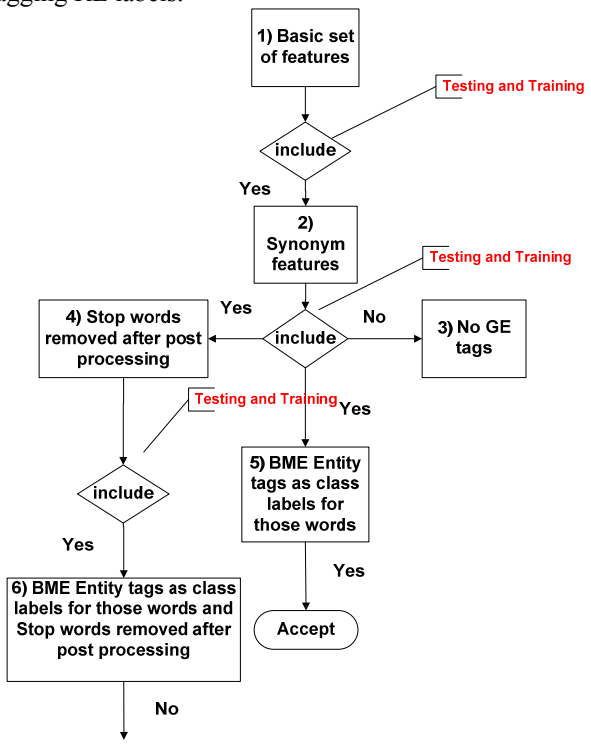

Figure 3. The process of experimenting with different combinations of features and class labels

We experimented with different combinations of features and classes, shown in Figure 3, and the results for each of them are explained in more details in Section 4. The list of experiments performed includes –

1. Experiment with the original feature set which includes all the features mentioned in Section 3.3.
2. Experiment with trigger words gazetteer with synonym features.
3. Experiment with removing stop words after assigning contextual features to the data set.
4. Experiment with learning different sets of class labels, which include – 1) no GE class labels, such that all the gene names get their chunk tags as their class labels, and 2) all the biomedical entity words get the biomedical entity (BME) tags assigned by the GENIA tagger as their class labels and hence no GE class labels but three other class labels – B-protein, B-DNA, and B-RNA.
5. Experiment with different combinations of Experiments 3 and 4 settings.

Once we decide on an appropriate set of features and class labels, we perform training and testing on our larger data set. The comparison results are presented in the next section.

4. Results

In this section we present the experimental results of our model in detecting the coexpression predicates from text using DCRFs. We started with 500 full length papers in our experiment and performed pre-filtering on it. After down sampling we were left with approximately 15,010 sentences all together. As we used a supervised learning approach, each word in each of these sentences was manually tagged as the ground truth for learning. Out of the total 15,010 sentences we have around 3,130 positive sentences which are only 20% of the total data set. Once the sentences are tagged appropriately, we divide the whole data set into training and testing files and perform training and testing on them using DCRFs. We present our results in the form of Precision, Recall and F-measure. Both Precision and Recall scores are combined into F-measure, which is a weighted harmonic mean of the two. F1-measure gives equal weights to both Precision and Recall, whereas F2-measure gives higher weight to Recall.

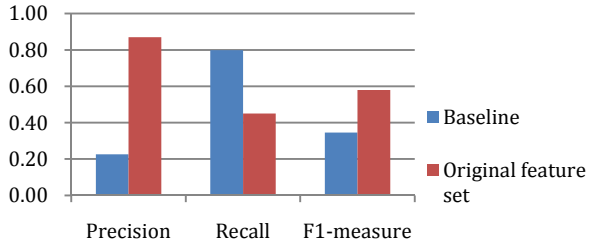

Figure 4. Comparison of results obtained by baseline and that of the model trained with the original set of features

Figure 4 above shows a comparison between the results of our **baseline model** and that of model trained with the **original set of features**. In the baseline experiment we perform the basic term matching from training to testing file. All the terms in the training files that correspond to coexpression terms were collected in a training corpus, and then each testing file was matched against it. If any word in the testing file matches some word in the training corpus, then the file is tagged as positive. This can also be considered as dictionary matching where the dictionary is built each time from a set of training files. The experiment with the original set of features involves experimenting with the set of features mentioned in Section 3.3. We can see that although baseline approach has a higher Recall score, it has a very poor Precision. This is because the baseline approach is solely based on keyword matching, and the contextual information is missing. One common example is the word "express" along with some other word like "together" or "both". This word, if used alone without a helping word like "both" or "together," does not indicate anything about coexpression relationships, but in the baseline approach every "expression" word will be tagged as positive in a test file. This will result in lots of false positives and lead to low Precision. Our model, trained with just the original set of features without feature selection, is almost 23% better than the baseline model in terms of F-measure.

As Figure 3 shows, we performed five additional sets of experiments to find the most appropriate set of features and class labels for this problem. Table 1 shows the 10 fold cross validation result of each of those experiments.

Table 1. Experimental results with different combinations of features and class labels

	Precision	Recall	F1	F2
Trigger words	0.81	0.51	0.62	0.55
No stop words	0.81	0.49	0.61	0.54
No GE labels	0.81	0.58	0.67	0.61
BME tags as labels	0.82	0.56	0.66	0.59
No stop words & BME tags	0.80	0.50	0.61	0.54

The first experiment involves experimenting with **Trigger words** and the results are presented in the second row of the table. Trigger words are the words that most commonly describe the variable that we want to classify. A gazetteer of these words can be created and a model can be tuned on them. We created a list of approximately 200 trigger words by including words like coexpress and their inflected forms. Whenever a word in a file matches some word in the trigger word gazetteer, it is assigned a feature referred to as "synonym feature," i.e., tagged with the word "coexpress". This improved the Recall of our system by 6%, which means that our model was able to identify more of the positive sentences. Since we achieved a significant improvement in the performance by adding this feature (as shown in Table 1), we included it for the rest of our experiments.

Stop words are the common occurring words like articles and prepositions that do not contain any useful information regarding the semantic content of the text. It is a common practice in natural language processing to filter out these words from the text as a preprocessing step. In the second trial we removed all the stop words from the training files but kept the contextual features for each remaining word. The result of this **No stop words** trial is presented in row 3 of Table 1. Although this approach performed better than the original feature set experiment, we did not achieve any significant improvement in this experiment (as shown in Table 1). This is probably because once these words were removed we also lost their syntactic tags/labels. And as we know DCRFs also learn the relationship between labels, thus losing those labels within a sentence may throw off the model and hence lead to poorer performance.

As aforementioned, DCRFs help in capturing complex relationship between labels and hence are useful in chunking task. In our task of relationship extraction, we want to identify the words describing coexpression as well as genes, and hence the performance of our DCRFs model is influenced by the prediction accuracy of GE as well as RE tags in a sentence. Therefore, we decided to further experiment with the class labels too. In this third experiment, mentioned as **No GE labels** in Table 1, we replaced all the GE class labels for gene names with their chunk tags 'B-NP' assigned by GENIA tagger. Hence in this case we made no distinction between the gene names and the other regular words in terms of the way their class labels are assigned. This leads to the least guessing work in model training and the prediction of class labels for non RE terms as there is a near-deterministic relationship between the input feature and the class label. This approach shows the highest improvement on our dataset, a 9% increase from the experiment with the original set of features, but does not give any specific information about the genes that are coexpressed. Consequently we did not include this in our final experiment.

Thinking along the same lines as before, we replaced all the GE class labels from the gene name words with the biomedical named-entity tags (not the same as chunk tags) that GENIA tagger assigns to those words. We also assigned those class labels to all the words that are tagged as biomedical entities by GENIA tagger. Hence not just the genes involved in the coexpression have those labels, but all of the gene names, protein names, and RNAs are assigned those labels. Though, our model now had three additional class labels to learn (B-protein, B-DNA and B-RNA), it still performed almost as well as the **No GE labels** experiment (see Table 1, row 5 **BME tags as labels**). However, with this approach, we not only get the

coexpression relationship words extracted but also the genes that are involved in the relationships. Recall of our system is also significantly higher, almost 11% higher than that of the experiment with the original set of features.

In the last test, we combined experiments 2 and 4, i.e., all the biomedical entities got their GENIA tagger assigned tags as the class labels and all the stop words were removed from sentences. However, this combination did not obtain any performance gain probably for the same reasons as stated earlier for experiment 2.

Finally, by analyzing all of these experiments on the development test set and training set, we came up with the following set of features and class labels for training our model. These include –

1) Local and contextual features as mentioned in Section 3.3.
2) Synonym features as used in experiment 1.
3) Class label RE for words that express coexpression relationships.
4) Class labels B-protein, B-DNA and B-RNA for tagging biomedical entities.

We can observe that although the Precision of the original model is higher, its Recall value is not as good as the others. There is always a trade-off between the Precision and Recall values, but it is essential to keep both of them as high as possible. We can see that the Recall values of those experiments with feature tuning, when compared with the original baseline feature set, have increased significantly more than the decrease in the Precision values. F2-measure gives more weight to Recall as the ability to retrieve more positive sentences is more emphasized in this specific application.

We also compared the performance of DCRFs with the baseline approach and Naïve Bayes, a well known classification algorithm. We used the Weka implementation of this Naïve Bayes and tested it with the same final set of features and class labels as was used for DCRFs [13]. Table 2 shows the comparison result of our approach with the baseline approach and Naïve Bayes approach.

Table 2. Comparison results with the baseline approach and another machine learning approach

	Precision	Recall	F1-measure
DCRFs	0.82	0.56	0.66
NaiveBayes	0.45	0.29	0.34
Baseline	0.23	0.80	0.35

5. Conclusions and future work

In this paper we present a framework for extracting predicates that state coexpression relationships among genes. We trained a DCRFs model based on semantic analysis of text to classify papers that talk about gene-gene coexpression relationships.

In the future we would like to investigate different grammatical parsing methods, like dependency parsing and use the parse trees generated as additional features to train our model. We believe that this can improve our results. Another direction we would like to explore is semi-supervised learning, as it requires less human effort of labeling the ground truth for learning, and may still give sufficiently high accuracy.

This work can be considered as a step towards the semantic based information extraction systems that can help researchers to extract accurate and relevant information from the vast array of textual information.

6. References

[1] I. Coulibaly and G.P. Page, "Bioinformatic tools for inferring functional information from plant microarray data II: analysis beyond single gene," *International Journal of Plant Genomics*, vol. 2008, Article ID 893941, 13 pages, 2008.

[2] L.F. Rau, P.S. Jacobs,. and U. Zernik, "Information extraction and text summarization using linguistic knowledge acquisition," *Information Processing & Management* 25 4, pp. 419–428, 1989.

[3] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," *in Proceedings of IEEE/WIC International Conference on Web Intelligence*, Halifax, Canada, pp. 702-705, 2003.

[4] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden markov model structure for information extraction," *in Proceedings of AAAI Workshop on Machine Learning for Information Extraction*, pp. 37-42, 1999.

[5] X. Wei, B. Croft, and A. McCallum, "Table extraction for answer retrieval," *Information Retrieval Journal*, 9 (5), pp. 589-611, 2006.

[6] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," *in Proceedings of the Seventh Conference on Natural Language Learning* (CoNLL), Edmonton, Canada, 2003.

[7] M. Bundschus, M. Dejori, M. Stetter, V. Tresp, and H.P. Kriegel, "Extraction of semantic biomedical relations from text using conditional random fields," *BMC Bioinformatics*, 9 (207), 2008.

[8] C. Sutton, K. Rohanimanesh, and A. McCallum. "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data," in *Proceedings of ICML*, Banff, Alberta, Canada, 2004.

[9] Apache PDFBox – Java PDF library, Available at http://incubator.apache.org/pdfbox/index.html

[10] GENIA Tagger 3.0, Available at www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger.

[11] C. Sutton, and A. McCallum, "An introduction to conditional random fields for relational learning," *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[12] C. Sutton, "GRMM: GRaphical Models in Mallet," Available at http://mallet.cs.umass.edu/grmm 2006.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, Volume 11, Issue 1, 2009.