




A Flexible Model for Enterprise Document Capturing Automation

Juris Rāts¹, Inguna Pede¹, Tatjana Rubina² and Gatis Vītols²

¹*RIX Technologies, Blaumana 5a-3, Riga, LV-1011, Latvia*

²*Faculty of Information Technologies, Latvia University of Life Sciences and Technologies,
2 Liela str., Jelgava, LV-3001, Latvia*

Keywords: Machine Learning, Document Classification, Enterprise Content Management, Python, Elasticsearch.

Abstract: The aim of the research is to create and evaluate a flexible model for document capturing that would employ machine learning to classify documents feeding them with values for one or more metadata items. Documents and classification metadata fields typical for Enterprise Content Management (ECM) systems are used in the research. The model comprises selection of classification methods, configuration of the methods hyperparameters and configuration of a number of other learning related parameters. The model provides user with visual means to analyse the classification outcomes and those to tune the further steps of the learning. A couple of challenges are addressed along the way – as informal and eventually changing criteria for document classification, and imbalanced data sets.

1 INTRODUCTION

The process of capturing new documents in an Enterprise Content Management (ECM) system includes document labelling - filling a number of metadata fields (like document type, folder, case, a user to route document to) with values. Received documents are routed to an employee responsible for document labelling who must read the document and understand the content to classify and label it properly. Although organizations usually have some guidelines or rules to guide this process, the proper labelling depends significantly on the persons experience and knowledge – usually hard to be explained in a set of formal rules.

A convenient method for handling this case is to use machine learning methods (as no formal algorithms can be easily specified). We use supervised machine learning as ECM normally has a set of properly labelled samples (documents already in the repository).


A large number of machine learning algorithms are developed and are ready for use. We aim to create in this research a flexible framework on top of the available algorithms that would provide users with:


- Classification robots that would learn to classify the captured documents;
- Advanced visual tools analysis of the results of document classification
- Configuration tools allowing to tune easily the next steps or robot learning and document classification.


Important advantage of our approach is that we rely mainly on machine learning that allows to reduce (although not fully eliminate) the manual work to create classification rules used in rule-based classification systems.

2 RELATED WORK

Text classification methods are researched by various authors in a number of fields, as sentiment analysis (Avinash M & Sivasankar E, 2019, Pahwa, Taruna, & Kasliwal, 2018, Fu, Qu, Huang, & Lu, 2018, Maas et al., 2016, Saif, Fernandez, He, & Alani, n.d., 2014, Tam Hoang, 2014, Pang & Lee, 2008), news classification (Kadriu, Abazi, & Abazi, 2019), web page classification (Shawon, Zuhori, Mahmud, & Rahman, 2018) etc. Some commercial mainly rule-

^a <https://orcid.org/0000-0002-3406-7540>

^b <https://orcid.org/0000-0002-1466-8031>

^c <https://orcid.org/0000-0002-4131-8635>

based solutions are created for similar tasks like Xtracta (“Xtracta: Automated Data Entry Software Powered by AI,” 2010), Serimag (“Serimag - Artificial Intelligence for document automation,” 2007), ABBY FlexiCapture (“Intelligent Document Processing Platform - ABBYY FlexiCapture,” 2019).

Document classification research frequently is based on machine learning methods, some of important methods being Naïve Bayes classifier, K-Nearest Neighbours, Support Vector Machine and Deep learning models (Kowsari et al., 2019).

Machine learning methods operate numeric data therefore text documents have to be translated to numeric vectors (i.e. vectorized) to employ them. One of the core approaches to the text vectorization is the Bag of Words (BoW) model. BoW has been successfully applied in broad range of domains as the medical domain (Lauren, Qu, Zhang, & Lendasse, 2018) with automating medical image annotation (Lauren et al., 2018), biomedical concept extraction (Dinh & Tamine, 2012), and recommender systems for medical events (Bayyapu & Dolog, 2010).

BoW ignores the semantic information therefore several extensions of the approach are developed. This includes Bag of meta-words (BoMW) (Fu et al., 2018) that uses meta-words instead of words as building blocks.

Various researchers (Kadriu et al., 2019), (Stein, Jaques, & Valiati, 2018) have analysed usage of embedding methods (Word2Vec, GloVe, FastText) and found they can improve predicting accuracy in some cases (e.g. large data volumes) and make the learning curve steeper. Another research (Avinash M & Sivasankar E, 2019) compares tf-idf and Doc2Vec and shows Doc2Vec has better accuracy than tf-idf for most cases.

ECM systems normally handle large sets of voluminous documents which means the feature sets extracted for machine learning tend to have large dimensionality. Dimensionality reduction methods like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and non-negative matrix factorization (NMF) allow for more efficient use of machine learning algorithms because of the time and memory complexity reduction (Kadhim, 2019; Kowsari et al., 2019; Lauren et al., 2018).

Advantages and disadvantages of various machine learning methods are analysed by a range of researchers. Naive Bayes (NB) (Avinash M & Sivasankar E, 2019; Porter, 2006) shows good results in document classification. Kadhim (Kadhim, 2019) argues NB showed the second-best accuracy out of the 5 methods tested.

Support Vector Machine (SVM) is one of the most efficient machine learning algorithms (Karamizadeh, Abdullah, Halimi, Shayan, & Rajabi, 2014), applied in particular to text categorization. Several drawbacks exist though as the lack of transparency in results caused by a high number of dimensions.

Deep learning networks are applied for text classification, e.g. Recurrent neural networks, Convolutional neural network, Hierarchical Attention networks (Jacovi, Sar Shalom, & Goldberg, 2019; Johnson & Zhang, 2014; LeCun, Bengio, & Hinton, 2015; Yang et al., n.d.), as well as combined RCNN - Convolutional neural with Recurrent neural network (Lin, Fu, Mao, Wei, & Li, 2019). Basic methods as Naïve Bayes show good results with smaller data sets, while Convolutional neural network shows superior performance with larger data sets (Wei, Qin, Ye, & Zhao, 2018).

Top languages used for implementation of machine learning models are Python, Java and R. Python is the most popular in practical implementations (Ciapetti et al., 2019). A number of frameworks are available for development of machine learning solutions most popular being TensorFlow and PyTorch.

3 LEARNING MODEL

Machine learning algorithms may be grouped by learning style (supervised learning, unsupervised learning, semi-supervised learning) and by function (e.g. classification and regression). We use supervised-learning based classification algorithms in our research as ECM has large sets of labelled documents - ones already captured in the system and labelled by the users.

Basic concepts of the supervised machine learning based classification process are listed in Table 1.

Table 1: Basic concepts.

Concept	Explanation
Training/test sample	A structure consisting of a feature set and a label. Training samples are used for learning by the machine learning algorithm, test samples are used to evaluate the performance of the learning.
Feature set	A set of (numeric) values representing the training sample
Label	A category having two or more values alias classes. The label class is what the classification algorithm has to predict.

Machine learning based document classification systems generally consist of four phases: feature and label extraction, dimension reduction, classifier selection (and applying), and evaluations (adapted from (Kowsari et al., 2019)).

Feature extraction is arguably one of the most important factors in successful machine learning projects (Faggella, 2019). Feature extraction deals with converting the sample text to a set of (numeric) features usable for machine learning algorithms.

Dimension reduction is an optional phase that aims to reduce the dimensionality of the feature set used for training.

Evaluation. The performance of the trained algorithm is evaluated on the test data before to use it for prediction. Evaluation results may be used to select the best performing algorithms (or to decide if the algorithm has been trained sufficiently).

Frequently it is not possible to say in advance which of the classification algorithms will perform better for a domain in question. The convenient approaches are to use several algorithms for training and to select the best performing, or use ensemble methods to combine the predictions of individual algorithms.

Some more aspects have to be addressed before the learning model can be applied for the real-life classification task like ECM document classification.

3.1 Channels

A document of the ECM system may have several capturing channels (or simply *channels*). Sample channel could be a particular e-mail address of the organization, interface with an external application or interaction with a particular group of users who add documents manually. Organizations normally have several channels (several e-mail addresses, several streams of documents from external applications etc.) and they use channels to capture different kinds of documents (e.g. some e-mail address is used to capture invoices, other – to capture customer complaints). Thus, document classification model has to comprise several robots – each for its own channel. The robot must be trained on documents (and labels) captured earlier through the same channel.

The robot of the particular channel should profit from the rules existing in the organization that might link a channel to specific document metadata values. For example – documents captured through a particular e-mail address might be saved in a specific folder which means the folder is known and must not be predicted by the robot.

3.2 Continuous Training

Machine learning algorithms usually are evaluated on publicly available data sets. The evaluation consists of two steps:

- The algorithm is trained on training samples and tested on test samples;
- the algorithm is evaluated on separate sample set to make sure how it performs on new data.

In an ECM repository new data is captured constantly therefore a classification algorithm has to be trained repeatedly as new data appears. The classification model has to provide means to control a number of parameters, like frequency to train the classificatory, volume of training samples to use, *grace period* (recent time period when the data from the repository is not used yet for training as prone to errors, e.g. incorrect or missing label values) etc.

3.3 Label Extraction

ECM document classification usually involves several metadata fields to be determined (e.g. document type, folder, assignee). At least two approaches may be considered to handle this:

- Document may be classified separately for each of the fields;
- The label to predict may be related to a combination of all metadata fields (e.g. a separate label would be created for combination of document type, folder and assignee); the document is classified against the combined label.

Our experiments show the second approach is preferable as it takes less processing time (processing of each separate label takes as much time as the processing of the combined label) and the predicting accuracy for both cases is comparable.

The combined label case shows our model must handle conversion from metadata fields to labels and vice versa. Metadata values of the document must be converted to labels when preparing the learning data set. Labels predicted by the classification robot have to be converted back to metadata fields.

3.4 Rules

Document classification rules may change. E.g. the documents previously routed to employee A are switched to employee B. These types of problems have to be addressed outside the main learning process. A set of substitution rules may be introduced. For a sample above the rules may be introduced that relate employees to roles (responsible for a certain

document category). If the employee roles change the substitution rules are changed accordingly.

3.5 Handling Imbalanced Data

The analysis of the data we use for the current research shows that the top five most popular labels (combined label document type + document folder + case) account for 94.76% of all samples while the rest (43) - for only 5.24%. The similar situation is identified in number of other ECM data sets we explored. The said means we are dealing with highly imbalanced data sets.

Imbalanced data sets can cause a number of problems for classification algorithms (Wong, Kamel, Sun, & Wong, 2011) for domains where rare cases tend to be more significant than the frequent ones (e.g. disease diagnosis and fraud detection). This does not apply for our case.

The problem we have though is that rare cases have less training data. We propose the method of forceful Balancing of Major Classes (BMC) to improve the learning data. BMC consists of four consecutive steps.

1. The set of N label classes L is ordered descending by appearance count in samples (popularity) into ordered set L^{ord} .
2. L^{ord} is split into two sets – major classes (or Majors - M_n , comprising first n classes from L^{ord}

$$M_n = (L_1^{ord}, L_2^{ord}, \dots, L_n^{ord}) \quad (1)$$

and minors

$$m_n = (L_{n+1}^{ord}, L_{n+2}^{ord}, \dots) \quad (2)$$

with total popularity

$$P(m_n) = \sum_{i=n+1}^N L_i^{ord} \quad (3)$$

not more than some a priori set *Minors Threshold* (T_m , $P(m_n) \leq T_m$), such as $P(m_{n-1}) > T_m$.

3. Majors set is supplemented with Others class (O) that represents all minor classes

$$M_n = (L_1^{ord}, L_2^{ord}, \dots, L_n^{ord}, O) \quad (4)$$

4. Training sample set is created in equal volumes for each Major class.

Class O has a special meaning as predicting class O means that algorithm is not able to determine the meta data fields for the case. Therefore, in our model the classification algorithm has 3 possible prediction outcomes – accurate prediction, false prediction and

no prediction. It should be noted that for a number of domains no prediction may be a better result than a false prediction.

The proposed BMC method allows to produce balanced set of training data for highly imbalanced data sets as long as there are enough samples in a total data repository. For extremely rare classes this may not be the case therefore introducing “no prediction” feature has a good ground as it can be used to configure the learning process to abandon predicting rare cases where there are no sufficient data anyway.

4 THE DOCUMENT CLASSIFICATION MODEL

As noted before there is no ground to differentiate the importance of label classes for the research domain in question (ECM document classification). The main metric to measure the performance of the classification against is the prediction accuracy on all samples in total. Considering the learning model proposed in the previous chapter the goal of the document classification model may be to maximize true predictions or to minimize false predictions. The appropriate goal has to be set by user.

Other requirements of the proposed document classification model are:

- the model handles continuous learning process as described in the previous chapter; the robots are trained periodically considering the new data and the user is empowered with advanced visualisation tools to analyse the samples classified incorrectly by the robots and to eventually change the configuration for the next training cycles;
- the model supports multiple document classification robots, each for its own channel of document capturing;
- the model supports user defined rules for conversion between meta fields and labels;
- the model supports a broad range of parameters, including frequency of robot retraining, grace period, sample volumes, feature extraction and classification methods (includes selection of methods and configuration of methods parameters), Minors Threshold etc.

The document classification model we propose comprises three processes (sample retrieval, training and predicting) that act on three data stores (Figure 1).

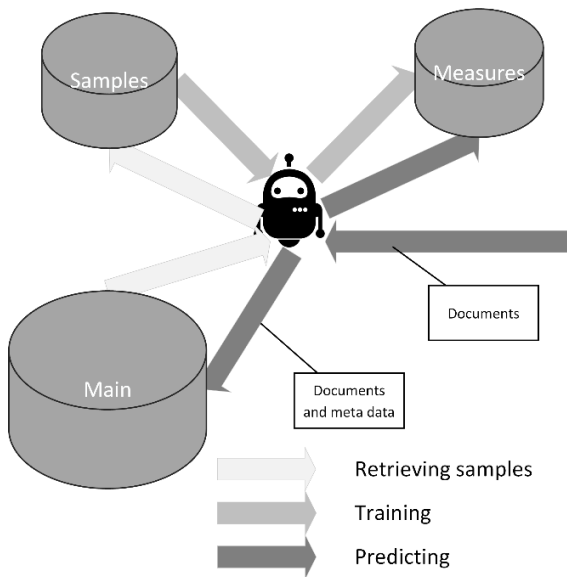


Figure 1: Document classification.

Sample retrieval process is executed periodically to retrieve new (and modified) samples (documents and meta data) from the main store (ECM systems database) and save them into the Samples store. Samples are tagged with channels they relate to.

Table 2: Model parameters.

Parameter	Comments
Vectorization methods	Tfidf, hashing
Classification methods	Stochastic gradient descent, Multinomial naïve Bayes, Passive-aggressive classifier, Logistic regression, Support vector classifier, Linear support vector classifier, Convolutional neuron networks
Sample volume	Count of samples (training plus testing) used to train a robot
Minors threshold	Determines the relative part of the samples related to the label classes not included in the Majors set (e.g. for Minors threshold 0.1 not more than 10% of all samples should relate to Minor classes).
Batch handling	Determines if robot training is performed in minibatches or in one go.
Test size	Determines the part of all samples hold out for testing.
Stop words	A list of words to ignore when vectorizing the text.

Training process retrieves (periodically) the samples for each robot, trains the robots and evaluates them. Statistics for robot training and evaluation are saved in the Measures store.

Trained robots listen for the respective channels and *predict* the metadata values, results are saved in the Measures store.

Important parameters supported by the model are listed in Table 2.

5 PERFORMANCE EVALUATION

The performance of the learning model was evaluated on a data set containing more than 160 thousand of documents (half of them digitalised). Combined label (document type + document folder + case) is used for the classification.

Seven classification methods and two vectorisation (converting text to numeric vectors) methods were tested for a number of hyperparameter values. Other parameters explored are volume of the training set, number of features selected for text representation, analysis of bigrams and trigrams, learning in minibatches (partial fit) vs learning in one go, Minors Threshold etc.

Figure 2 demonstrates influence of Minors Threshold. Rightmost bar (value 0) represents the case when robot attempts to predict values of all label classes while the leftmost (value 0.2) – when robot ignores (does not predict) the minority classes accounting in total for 20% of all samples.

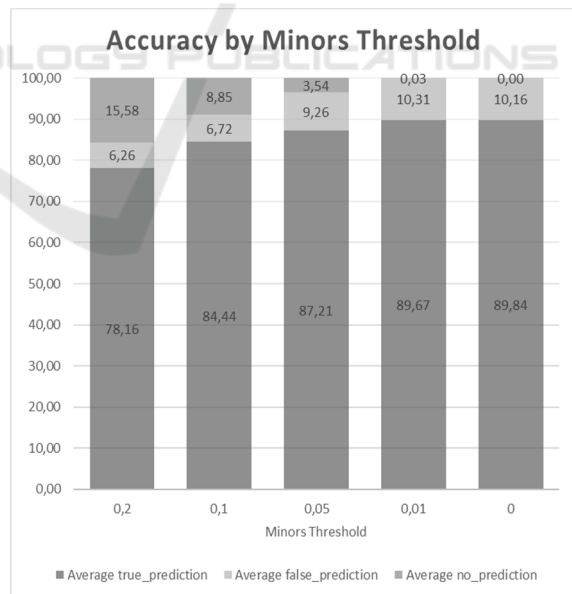


Figure 2: Accuracy by Minors Threshold.

The most significant conclusion from the test measurements though is that the accuracy of the document classification robots trained with different

classification methods do not differ significantly (see Figure 3 for sample comparison). Multinomial Naïve Bayes (MultinomialNB) has lower accuracy here but this may change with new documents captured. Every next learning cycle may have its own best methods. This means the most convenient approach to implement the model would be to create a flexible framework that allows for easy configuration of the learning process both when deploying and when maintaining the process in production. The framework should provide means for analysis of false and no predictions and the tools for easy configuration of all involved parameters.

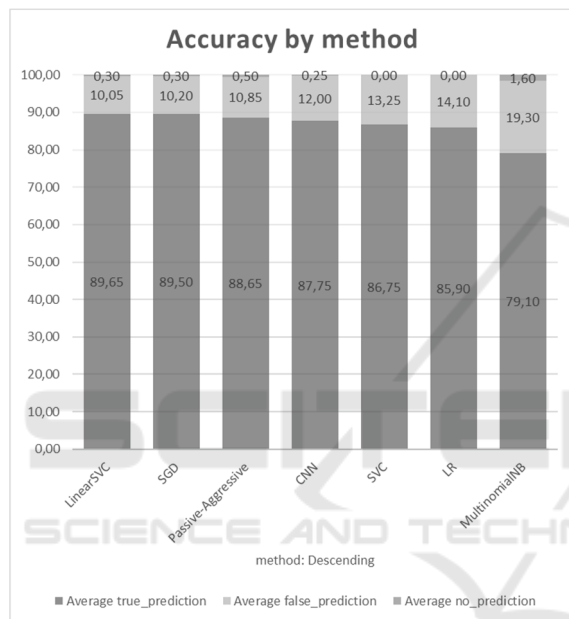


Figure 3: Accuracy by method.

Interestingly enough the deep learning methods like multi-layered Convolution Neuron Network (CNN) did not show better results than basic methods like Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), Linear SVC, Passive-Aggressive Classifier and Logistic Regression (LR).

Two more data sets have been tested for comparison. The results were similar to the ones revealed above.

6 FUTURE WORK

The prototype for document classification framework is currently in development. Elasticsearch is used to implement data stores for Samples and Measures, Kibana provides advanced visualisation means for data and performance analysis. Python (including

machine learning libraries sklearn and keras) is used to implement the functionality.

The results of our research support the conclusion made by several authors (e.g. Faggella, 2019) that the feature extraction is the most important phase of the machine learning based document classification process. The means should be incorporated into our model to allow further configuration of the feature extraction. Ontology based feature extraction (Kolle, Bhagat, Zade, Dand, & Lifna, 2018) is one of possible direction to proceed.

7 CONCLUSIONS

The proposed document classification model is based on the following main observations from the domain of research:

- Documents are captured by ECM system through channels, each channel handles its own specific set of documents;
- Label class distribution of a document set is highly imbalanced;
- The classification rules are unformal and subject to change.

Machine learning based document classification process has to be configured to work properly. The configuration includes selection of classification and vectorisation methods, tuning of hyperparameters of said methods, configuration of a number of parameters important for the learning process. The measurements we run did not reveal any significant advantages of any particular classification method. We have a ground to believe, in contrary, that methods and hyperparameters should be selected for the particular case (e.g. ECM document management process). Moreover – it is necessary to periodically analyse the performance of the classification model and to tune the configuration while in production to compensate for changes of the classification rules.

The said above means that the document classification system has to support tools both for analysis of its performance and for periodic fine tuning.

The experiments with the model demonstrated as well the superior importance of the feature extraction process for improving the document classification accuracy.

ACKNOWLEDGEMENTS

The research accounted for in this paper is co-funded by the European Regional Development Fund (ERDF) (project No. 1.2.1.1/18/A/003).

REFERENCES

- Avinash M., & Sivasankar E. (2019). *A Study of Feature Extraction techniques for Sentiment Analysis*. 1–12.
- Bayyapu, K. R., & Dolog, P. (2010). Tag and Neighbour Based Recommender System for Medical Events. *Proceedings of the First International Workshop on Web Science and Information Exchange in the Medical Web, MedEx 2010*, 14–24. APA.
- Ciapetti, A., Florio, R. Di, Lomasto, L., Miscione, G., Ruggiero, G., & Toti, D. (2019). *NETHIC: A System for Automatic Text Classification using Neural Networks and Hierarchical Taxonomies*. 296–306.
- Dinh, D., & Tamine, L. (2012). Towards a context sensitive approach to searching information based on domain specific knowledge sources. *Journal of Web Semantics*, 12–13, 41–52. <https://doi.org/10.1016/J.WEBSEM.2011.11.009>
- Faggella, D. (2019). What is Machine Learning? Retrieved October 10, 2019, from <https://emerj.com/ai-glossary-terms/what-is-machine-learning/>
- Fu, M., Qu, H., Huang, L., & Lu, L. (2018). Bag of meta-words: A novel method to represent document for the sentiment classification. *Expert Systems with Applications*, 113, 33–43. <https://doi.org/10.1016/J.ESWA.2018.06.052>
- Intelligent Document Processing Platform - ABBYY FlexiCapture. (2019).
- Jacovi, A., Sar Shalom, O., & Goldberg, Y. (2019). *Understanding Convolutional Neural Networks for Text Classification*. 56–65. <https://doi.org/10.18653/v1/w18-5408>
- Johnson, R., & Zhang, T. (2014). *Effective Use of Word Order for Text Categorization with Convolutional Neural Networks*.
- Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, 52(1), 273–292. <https://doi.org/10.1007/s10462-018-09677-1>
- Kadriu, A., Abazi, L., & Abazi, H. (2019). Albanian Text Classification: Bag of Words Model and Word Analogies. *Business Systems Research Journal*, 10(1), 74–87. <https://doi.org/10.2478/bsrj-2019-0006>
- Karamizadeh, S., Abdullah, S. M., Halimi, M., Shayan, J., & Rajabi, M. J. (2014). Advantage and drawback of support vector machine functionality. *I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings*, 63–65. <https://doi.org/10.1109/I4CT.2014.6914146>
- Kolle, P., Bhagat, S., Zade, S., Dand, B., & Lifna, C. S. (2018). Ontology based Domain Dictionary. *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 1–4. <https://doi.org/10.1109/ICSCET.2018.8537346>
- Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text Classification Algorithms: A Survey. *Information (Switzerland)*, 10(4). <https://doi.org/10.3390/info10040150>
- Lauren, P., Qu, G., Zhang, F., & Lendasse, A. (2018). Discriminant document embeddings with an extreme learning machine for classifying clinical narratives. *Neurocomputing*, 277, 129–138. <https://doi.org/10.1016/J.NEUCOM.2017.01.117>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lin, R., Fu, C., Mao, C., Wei, J., & Li, J. (2019). *Academic News Text Classification Model Based on Attention Mechanism and RCNN*. https://doi.org/10.1007/978-981-13-3044-5_38
- Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2016). Learning Word Vectors for Sentiment Analysis. *European Review for Medical and Pharmacological Sciences*, (January 2011), 9. <https://doi.org/10.1155/2015/915087>
- Pahwa, B., Taruna, S., & Kasliwal, N. (2018). Sentiment Analysis- Strategy for Text Pre-Processing. *International Journal of Computer Applications*, 180(34), 15–18. <https://doi.org/10.5120/ijca2018916865>
- Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis: Foundations and Trends in Information Retrieval*. 2(1–2), 1–135. <https://doi.org/10.1561/15000000011>
- Porter, M. F. (2006). An algorithm for suffix stripping. *Program*, 40(3), 211–218. <https://doi.org/10.1108/00330330610681286>
- Saif, H., Fernandez, M., He, Y., & Alani, H. (n.d.). *On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter*.
- Serimag - Artificial Intelligence for document automation. (2007).
- Shawon, A., Zuhori, S. T., Mahmud, F., & Rahman, J. (2018). Website Classification Using Word Based Multiple N-Gram Models And Random Search Oriented Feature Parameters. *2018 21st International Conference of Computer and Information Technology (ICCIT)*, (21-23 December), 1–6. <https://doi.org/10.1109/ICCITECHN.2018.8631907>
- Stein, R. A., Jaques, P. A., & Valiati, J. F. (2018). *An Analysis of Hierarchical Text Classification Using Word Embeddings*. <https://doi.org/10.1016/j.ins.2018.09.001>
- Tam Hoang, D. (2014). *Sentiment Analysis: Polarity Dataset*. Charles University in Prague.
- Wei, F., Qin, H., Ye, S., & Zhao, H. (2018). Empirical Study of Deep Learning for Text Classification in Legal Document Review. *2018 IEEE International*

- Conference on Big Data (Big Data)*, 3317–3320.
<https://doi.org/10.1109/BigData.2018.8622157>
- Wong, A., Kamel, M. S., Sun, Y., & Wong, A. K. C. (2011).
Classification of imbalanced data: a review Pattern-
Directed Aligned Pattern Clustering View project
Pattern discovery in gene expression data View project
CLASSIFICATION OF IMBALANCED DATA: A
REVIEW. *Article in International Journal of Pattern
Recognition and Artificial Intelligence*, 23(4).
<https://doi.org/10.1142/S0218001409007326>
- Xtracta: Automated Data Entry Software Powered by AI.
(2010).
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy,
E. (n.d.). *Hierarchical Attention Networks for
Document Classification*. Retrieved from
<https://www.cs.cmu.edu/~diyiy/docs/naacl16.pdf>

