International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:14, No:11, 2020

Developing an Advanced Algorithm Capable of Classifying News, Articles and Other Textual Documents Using Text Mining Techniques

R. B. Knudsen, O. T. Rasmussen, R. A. Alphinas

***Abstract*—**The reason for conducting this research is to develop an algorithm that is capable of classifying news articles from the automobile industry, according to the competitive actions that they entail, with the use of Text Mining (TM) methods. It is needed to test how to properly preprocess the data for this research by preparing pipelines which fits each algorithm the best. The pipelines are tested along with nine different classification algorithms in the realm of regression, support vector machines, and neural networks. Preliminary testing for identifying the optimal pipelines and algorithms resulted in the selection of two algorithms with two different pipelines. The two algorithms are Logistic Regression (LR) and Artificial Neural Network (ANN). These algorithms are optimized further, where several parameters of each algorithm are tested. The best result is achieved with the ANN. The final model yields an accuracy of 0.79, a precision of 0.80, a recall of 0.78, and an F1 score of 0.76. By removing three of the classes that created noise, the final algorithm is capable of reaching an accuracy of 94%.

***Keywords*—**Artificial neural network, competitive dynamics, logistic regression, text classification, text mining.

I. Introduction

GLOBAL firms often engage each other in multiple distinct markets. Banks have more than one branch and their activities usually span over several markets, placing them in competitive relationships with other banks; airline companies compete for passengers on multiple routes around the world; Michelin and Goodyear battle for global dominance in the tire market, as Pepsi and Coca Cola fight to be the most recognizable soda brand in the world. This is often described as Multimarket Competition, which is broadly defined as situations where companies compete against the same rival, simultaneously on different markets [1]-[3].

Competitive dynamics have in recent years gained substantial attention from researchers working with global strategic actions and reactions. The multinational domain has in years been a focus where Multinational Enterprises (MNEs) and their global actions have been the focal research objective [3]-[5]. MNEs strive for global competitive advantages and they, therefore, respond to rivals' actions to improve or defend their current and future market position. Findings in articles resonate that businesses are becoming more and more global which thereby requires them to compete with highly integrated strategies [3]-[6]. In the domain of applying these global strategies, the response time has been in focus. MNEs must constantly respond to their competitors in order to defend or improve their market position and competitive advantages and it is therefore important for enterprises to know how their rivals act and react with regard to its rivals. The response time is, therefore, a key indicator for a firm's performance on the multi-domestic strategic level [3].

An example presented in literature is that enterprises have a faster response time to actions that are permanent and irreversible, to actions from rivals with a high amount of resources and similar profiles and to actions taken in strategically important markets [3], [7]. A result of the competitive dynamics research is 11 types of competitive actions companies can take when working on a global scale with multinational strategies [3]-[6]. The 11 competitive actions are: Pricing actions, Major Product Action, Marketing Action, Minor Product Action, Capacity Action, Change in Structure and Systems, Distribution and Service Improvement, Entry into New Market Niche, International Expansion, New Product Action and Technology Innovation. It is important to be able to identify these competitive dynamics and know how to read them.

According to [8], the words that people use in narrative texts are a reflection of their thoughts. Therefore the words that are being used in narrative texts about a firm's activities, such as annual reports and news articles, can give away information about a firm's strategies and perceptions [8]-[10]. A method to capture these insights is TM. This describes the process of discovering patterns of interest in databases and using them for decision making. TM is an area that is capable of providing a significant competitive advantage to companies, and the interest and importance of the subject are, therefore, growing increasingly [11]. Multimarket competition and TM are both large fields of research. In the area of multimarket competition, it is seen that response time is a large field of interest, and the utilization of textual information can be relevant for businesses to gain competitive advantages. According to data mining, the possibilities of automation are capable of complementing the need for competitive dynamics research. It is therefore highly relevant to research this subject.

II. Data

In order to create an algorithm capable of classifying competitive actions, data from the global automobile industry

R. B. Knudsen and O. T. Rasmussen are Technology based business development engineers, 7400 Herning Denmark (e-mail: 201403742@post.au.dk, 201509477@post.au.dk).

R. A. Alphinas is with the Aarhus University, Department of Business Development and Technology, Birk Centerpark 15, 7400 Herning, Denmark (e-mail: roal@btech.au.dk).

are used as the case. The global automobile industry is deemed as a good setting for several reasons. First of all, the global automobile industry is known for having a high level of rivalry between the different automobile firms. Second, the automobile industry has a manageable amount of actors that are easy to identify. As the industry is largely oligopolistic, there is a high degree of strategic independence between the different firms. This means that an action performed by one of the firms is likely to have significant implications on competitors. Third, it is easy to identify distinct markets and the competitors that act upon it as there is a high degree of heterogeneousness in market characteristics around the world. Fourth, the information that is available about the industry is easily accessible. Lastly, the majority of the firms in the automobile industry are focused on automobile manufacturing, this means that diversification is low [3]-[6].

The data used for this research are a sheet containing 1.121 news articles from the automobile industry with their belonging labels (competitive actions). There exist 11 different labels and each news article has a unique ID number. Not all articles contain a competitive action and can therefore not be given a specific label. These articles without a label will, therefore, represent a new class called “no action”. This also prohibits the algorithm from forcing an article into one of the established competitive actions. There will, therefore, exist 12 classes in total. 100 “no action” articles are then added to the total number of articles making it a total of 1,221 news articles. These news articles will then lay the foundation for the training and tests of the algorithms. Fig. 1 shows the distribution of the provided data.

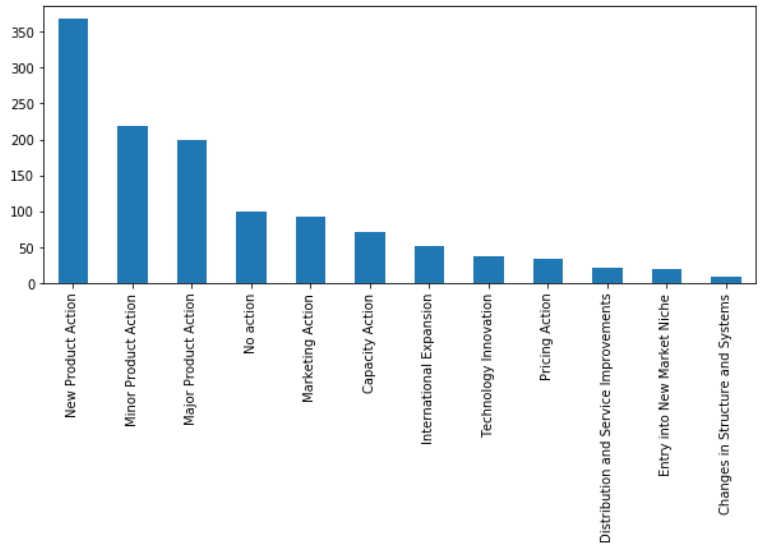

Fig. 1 Data distribution before augmentation

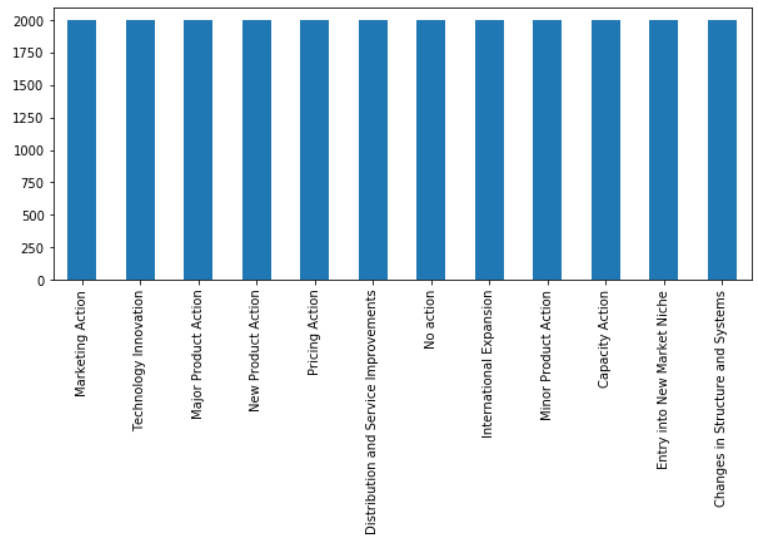

Fig. 2 Data distribution after augmentation

III. Theory

This section will contain the theoretical foundation of the research.

A. Data Augmentation

It is paramount to have a sufficient amount of data to create a classifier, as a classifier is often worthless without enough training data [12], [13]. In this research, a total number of 1,221 articles are used as the data set. This, however, is not a significantly large number when working with classification. As it might be too little data for the classifier to find any trends or patterns. Therefore, it is necessary to create new labeled data from the existing data. Further, there is a disproportionate amount of articles in each class, which can contribute to suboptimal predictions, as the algorithm has more "information" on some of the classes than others [12]. A method that is used in this research is the word shuffle method. The shuffling method is used in this research, in order to inflate the 1,221 original labeled articles, into a total of 24,000 after the augmentation. The numbers have been inflated, so there is a total of 2,000 articles of each class. The number of 2,000 is seen used in coding examples with similar objectives and this is therefore seen appropriate. Fig. 2 shows the distribution after augmentation of data.

B. Preprocessing Pipeline Creation

All data for this research paper have to be prepared before the training of the algorithms. Different pipelines are therefore made to identify the best preprocess for each algorithm. All of the pipelines will contain lower casing of words and the removal of punctuation. A total of eight pipelines are produced with varying configurations. These eight pipelines are then tested on nine different algorithms: Naïve Bayes, LR, Support Vector Machine, ANN, Convolutional Neural Network, Recurrent Neural Network and lastly, a Recurrent-Convolutional Neural Network. Of these, the best results are found with LR and ANN. It is seen that a pipeline consisting of only lower casing and removal of punctuation fitted the LR algorithm the best, and a pipeline containing lower casing, removal of punctuation, removal of stopwords, removal of common word, removal of rare words and the utilization of lemmatization fitted the ANN the best yielding the best results.

C. Evaluation Criteria

The first evaluation criterion is the accuracy. Accuracy is a measurement that shows how many correct predictions the model has predicted. Accuracy is measured between 0 and 1, which corresponds to a percentage where 0 is 0% and 1 is 100%. This means that the closer the accuracy is to 1, the better the model is at predicting, and therefore it is always strived to get as close to 1 as possible [13]. Accuracy in this research will measure the accuracy of the model to predict the correct class in which a specific article should be placed. Accuracy is therefore seen as a good parameter for measuring if the model is capable of predicting the correct outcome [13]. The calculation of the accuracy measure is made using the True Positive, True Negative, False Positive, and False Negative measures. The calculation can be seen in (1):

$$Accuracy = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

The second evaluation criterion is Precision and Recall. To understand Precision and Recall, one must first understand the confusion matrix. The confusion matrix can be used as a summary of the results of a prediction made on a classification problem. The number of correct and incorrect predictions the model has made is summarized and broken down by each class. The confusion matrix enables one to see in which areas that are confusing the classification model [13] while the confusion matrix is a great tool to gain an overview of where the model is having trouble predicting. It can be interesting to look at how accurate the model is at predicting positives, this is called the Precision of the classifier [13]. The formula for precision can be seen in (2):

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

The total number of correctly classified positives is divided by the total number of predicted positives. A high Precision means that an example that is labeled as positive is positive. Precision is often used in conjunction with the metric Recall. Recall is the number of correctly caught positive instances [13]. The formula for recall can be seen in (3):

$$Recall = \frac{TP}{TP+FP} \quad (3)$$

The total number of correctly classified positives is divided by the total number of positives. A high Recall means that the class is correctly classified [13].

The third and final evaluation criterion is the F1 score. The F1 score is the combination of Precision and Recall into one single metric. F1 score can be explained as the harmonic mean of Precision and Recall. Harmonic mean gives more weight to lower values, contrary to regular mean that treats all values equally. This means that if the Precision and Recall have a high score, the F1 score for the classifier will also be high [13]. The formulation for F1 score can be seen in (4):

$$F_1 = \frac{TP}{TP+\frac{TP+FP}{2}} \quad (4)$$

The F1 score favors classification algorithms which have similar values of Precision and Recall. This is, however, not always what is wanted when developing a classifier if in some cases Precision is favored over Recall and likewise [13].

D. LR

LR classification is done based on a predictor, a linear prediction function that assigns the values of dependent variables with weights. This process may involve nonlinear operations. First the data input is provided with the features and the real classification values. Then weights are given to each of the associated features. Following this is a link

function that converts the provided data to a range of 0 to 1. This link function is in this case a sigmoid function. Following this, the cost can be computed with a cost function which enables evaluation of the model. The practical calculations of LR can be seen in the following. The data (X,Y) are given where X is a matrix containing values with m examples and n features and Y is a vector with m examples. The objective of training a LR model is for the model to predict and assign a class to future values. As the first step of LR, a matrix is created containing random weights. The weights are then multiplied with the features, see (5):

$$a = w_0 + w_1x_1 + w_2x_2 + \cdots w_nx_n \quad (5)$$

The output of the above equation is then passed to the link function. The link function is also called the sigmoid function. This is formulated as in (6):

$$\hat{y_i} = \frac{1}{(1+e^{-a})} \quad (6)$$

Equation (6) is then followed by the calculation of the cost for the iteration. It is then formulated as (7):

$$cost(w) = (\frac{-1}{m}) \sum_{i=1}^{i=m} y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i}) \quad (7)$$

The derivative of the cost function is then calculated to find the gradient where the weights now can be optimized to optimize the prediction model. The gradient can then be formulated as (8):

$$dw_j = \sum_{i=1}^{i=n} (\hat{y} - y) x_j^i \quad (8)$$

The model can then be updated running it for several iterations until convergence with the update equation which is formulated as (9):

$$w_i = w_j - (a * dw_j) \quad (9)$$

E. ANN

ANNs are similar to the human brain. They consist of neurons which are trained to recognize complex patterns, then to output a prediction. The n number of inputs can be seen as the amount of features in a dataset. The process of passing data through the ANN is called forward propagation [13]. As an example, the training of a perception will be used. When providing input for the perceptron, the input value x_i is multiplied with w_i which is known as weights. The weights represent the strength of each connection between neurons. The weights furthermore decide the amount of influence a given input has on the output. The larger the value of the weight, e.g. w_1 has a larger value than w_2, then the input x_1 will have a higher influence on the outcome compared to x_2 because of their weights. This can be formulated as (10):

$$\sum (x_1 * w_1) + (x_2 * w_2) + \cdots + (x_n * w_n) \quad (10)$$

The inputs and weights row vectors are therefore $x=[x_1,x_2,...x_n]$ and $w=[w_1,w_2,...w_n]$. Their dot product is then given in (11):

$$x \cdot w = (x_1 * w_1) + (x_2 * w_2) + \cdots + (x_n * w_n) \quad (11)$$

and the summation is then equal to the dot product of the vectors *x* and *w* (12):

$$\sum = x \cdot w \quad (12)$$

A bias *b* then has to be added to the equation. The result of this addition is then called *z*. Bias is also called offset, because it is used to move the activation function to the left or right according to what is required to get the output values. This can be formulated as (13):

$$z = x \cdot w + b \quad (13)$$

The value of *z* then has to be passed to an activation function. The purpose of adding activation functions is to introduce non-linearity into the equation which then affects the output of each neuron. Without an activation function, the ANN would work as a basic linear function. The activation function furthermore has an effect on the speed of learning for the ANN. As a standard, a perceptron has a binary step function, which can be formulated as (14):

$$f(x) = \begin{cases} 0 \; for \; x \leq 0 \\ 1 \; for \; x \geq 0 \end{cases} \quad (14)$$

The step function will be replaced with the sigmoid activation function. The sigmoid activation function is also known as a logistic function. It can be formulated as (15):

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} \quad (15)$$

For this research, the aim is to classify news articles into 12 classes. The research objective, therefore, lies in predicting the likelihood of a certain example to belong to a certain class. In the output layer, there will be one node for each of the classes. The softmax activation function can be used in such a case, as it is capable of turning numbers into vectors that represent the probability of the distribution. Softmax can be formulated as (16):

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (16)$$

IV. Optimization Tests

On each of the algorithms, a grid search will be performed. This is to identify the optimal configuration of hyperparameters to increase the capability for classification. There will, therefore, be created several variations of both algorithms to identify the best configuration.

The tests will all be performed sequentially, meaning for instance when the best dropout rate is found, this will be used

in the next series of tests. This means that the best performance in a series of tests will carry on to the next series of tests, leading to the most optimal composition when all of the tests are performed.

The LR test withholds the following:

- The C value
- The max_iter value
- Combination of penalty & solver

The ANN test withholds the following:

- Drop-out test
- Layer test
- Node tests
- Mix of nodes test
- Optimizer test
- Number of epochs

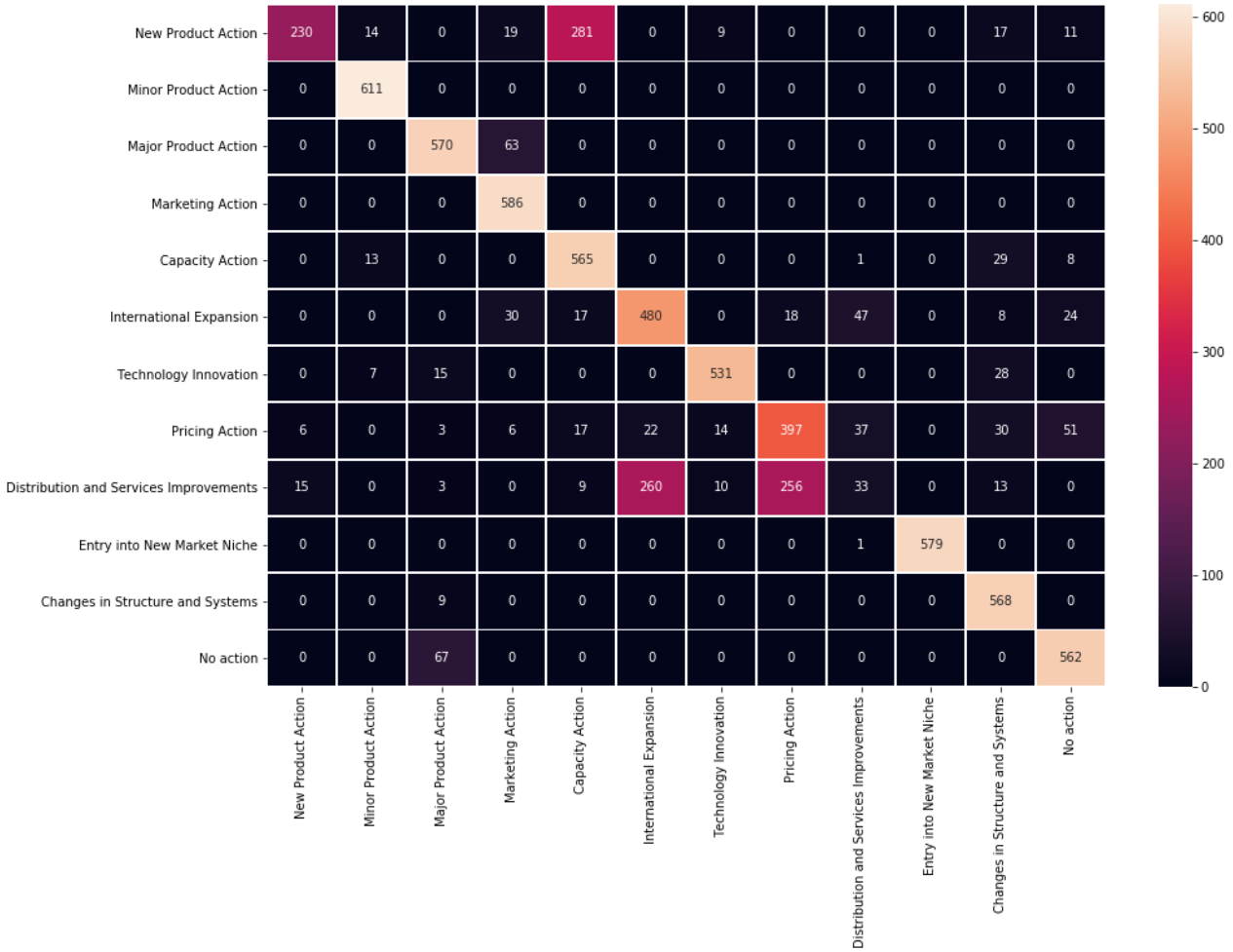

Fig. 3 Confusion Matrix for LR

V. Findings

After optimizing the parameters for each algorithm, the optimal combinations are identified.

For LR the optimal and final combination withholds a C value of 1000, a max_iter value of 100, and that the optimal penalty is L2 and the optimal solver is lbfgs (see Table I).

TABLE I
OPTIMAL PARAMETERS FOR LR

	C	Max_iter	Penalty	Solver
LR	1000	100	L2	lbfgs

With the above-mentioned parameters, the measurements in Table II are reached. The measurements are all the highest numbers identified above all tests.

TABLE II
MEASUREMENTS OF THE OPTIMAL LR

Accuracy	Precision	Recall	F1-score	AUC
0.793333	0.78	0.79	0.76	0.97865

The result of the ANN is seen to be a drop-out of 0.1, four layers without input and output layers each using 512 neurons. The optimal optimizer is seen to be Adam and the ANN in trained for five epochs. Final results and chosen parameters can be seen in Table III.

The final results of the chosen parameters can be seen in Table IV.

TABLE III
OPTIMAL PARAMETERS OF ANN

Dropout rate	Number of layers	Number of nodes	Combination of nodes	Optimizer	Number of epochs
0.1	4	512	512x4	Adam	5

TABLE IV
MEASUREMENTS OF THE OPTIMAL ANN

Accuracy	Precision	Recall	F1-score	Loss

As can be seen in Table V, the values of all the evaluation measurements are close. LR has a better accuracy and a higher recall where the ANN has a higher precision. The F1 score for both of the algorithms are 0.76. Due to the closeness of the measurements, it is seen needed to look further into the actual Precision, Recall and F1 score values.

TABLE V
COMPARISON OF LR AND ANN

	Accuracy	Precision	Recall	F1-score
LR	0.793333	0.78	0.79	0.76
ANN	0.792916	0.80	0.78	0.76

VI. DISCUSSION

When looking at the confusion matrices withholding the 7,200 unlabeled articles, in Figs. 3 and 4, for the two models, it can be seen that there are no significant variations.

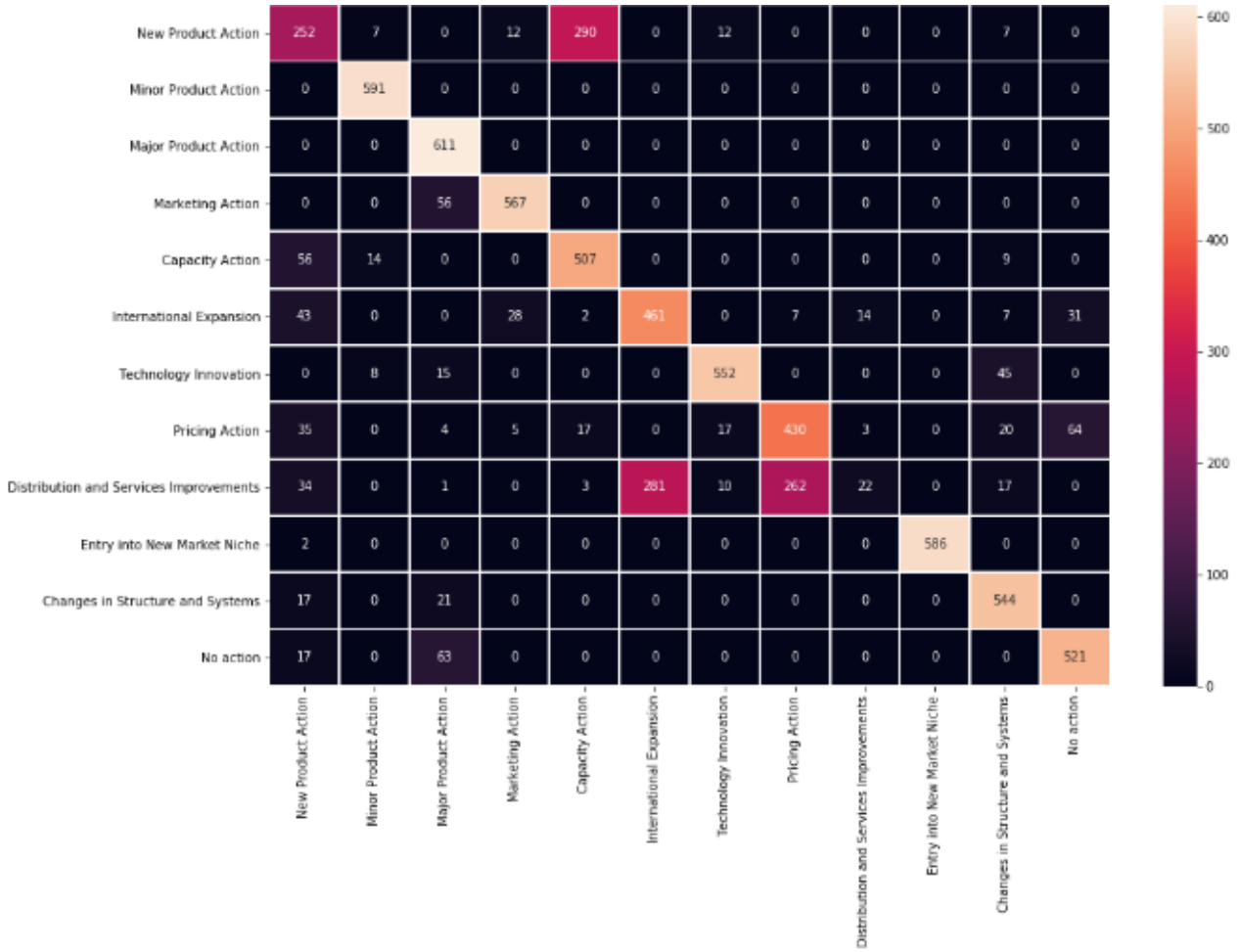

Fig. 4 Confusion Matrix for ANN

As can be seen in the two confusion matrices, the classification of the articles is very similar. Both of the models are having issues with classifying "New Product Action", "Capacity Action", "Distribution and Services Improvements", "International Expansion" and "Pricing Action" correctly. Due to this similarity, there is some indication of issues with the dataset with some or all of the mentioned classes.

When selecting between the LR model and the ANN model, it has to come down to small differences. Using the dataset with the 12 classes there is a small difference in accuracy of 0.00042, and the accuracy is therefore seen as mostly identical. However, in the Precision and Recall, the ANN is on average seen to have a 2% increase in Precision and only a 1% decrease in Recall compared to the LR. The ANN with its optimized parameter combination is therefore chosen as the

optimal classifier for solving the presented classification problem of this research. The ANN is the preferred model for this classification task due to generally preferred 2% more accurate predicted articles in trade off of 1% higher recall of news articles in the original dataset.

As a quick test, some of the problematic classes are removed. The Accuracy, Precision, Recall, and F1 score are here seen to have a significant increase. The three categories "New Product Action", "Capacity Action" and "Distribution and Services Improvements" are removed from the dataset to see how this would influence the performance outcome. Using both the LR model and the ANN model the results using the final models have improved.

Removing the three noisiest classes, the accuracy of the trained models are now reaching 94.15% for the LR model and 94.44% for the ANN model. This highly indicates that if some classes or noise from the data set are removed, the accuracy can reach high percentages. The Precision and Recall of both models are also increased significantly. For LR, the Precision before was 0.78 where it is reaching 0.94 after. The Recall likewise goes from 0.79 to 0.94. This is also happening for the ANN which previously had a Precision of 0.80 and 0.95 after, and a Recall before of 0.78 which is 0.94 after. It can, therefore, be seen that the accuracy measured with the original dataset had reached its limit for that specific dataset.

To get the higher measurements, it is therefore needed to clean and streamline the classes to make sure each article is only present in one class and possibly get more articles in the classes which do not have a high enough number of original articles before augmentation. By removing problematic classes, the new confusion matrices also seem to be cleaner and only the "Pricing Action" class is seen to have some minor difficulties with its recall.

As it can be seen in Figs. 5 and 6, the models are, even after the removal of some noisy classes, very close. They seem to have the same pattern in both measurements and in the looks of their confusion matrices. It can, therefore, be difficult to select which of the models would perform the best after a potential noise clean up and the addition of more articles to the training set.

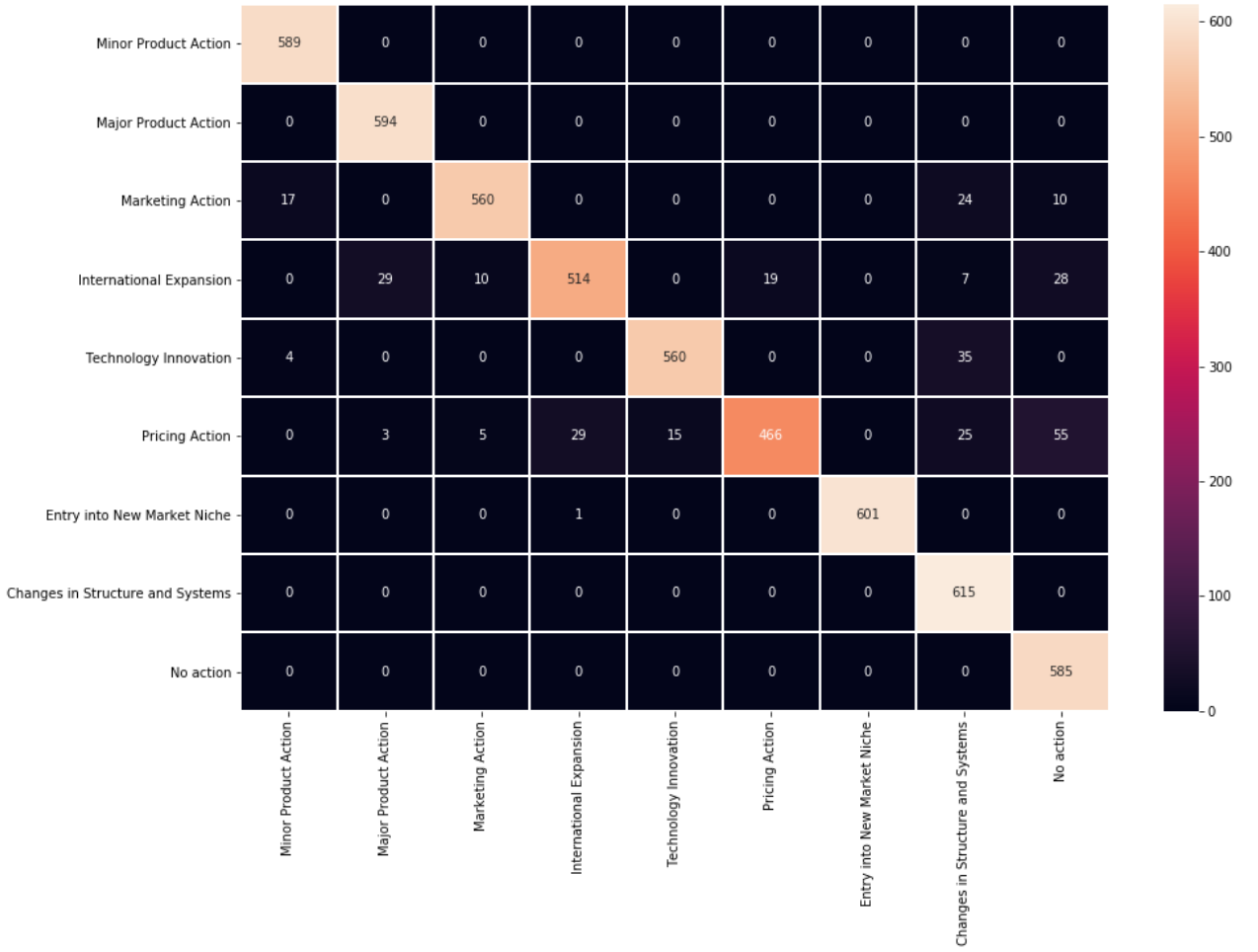

Fig. 5 Confusion Matrix for LR after removing classes

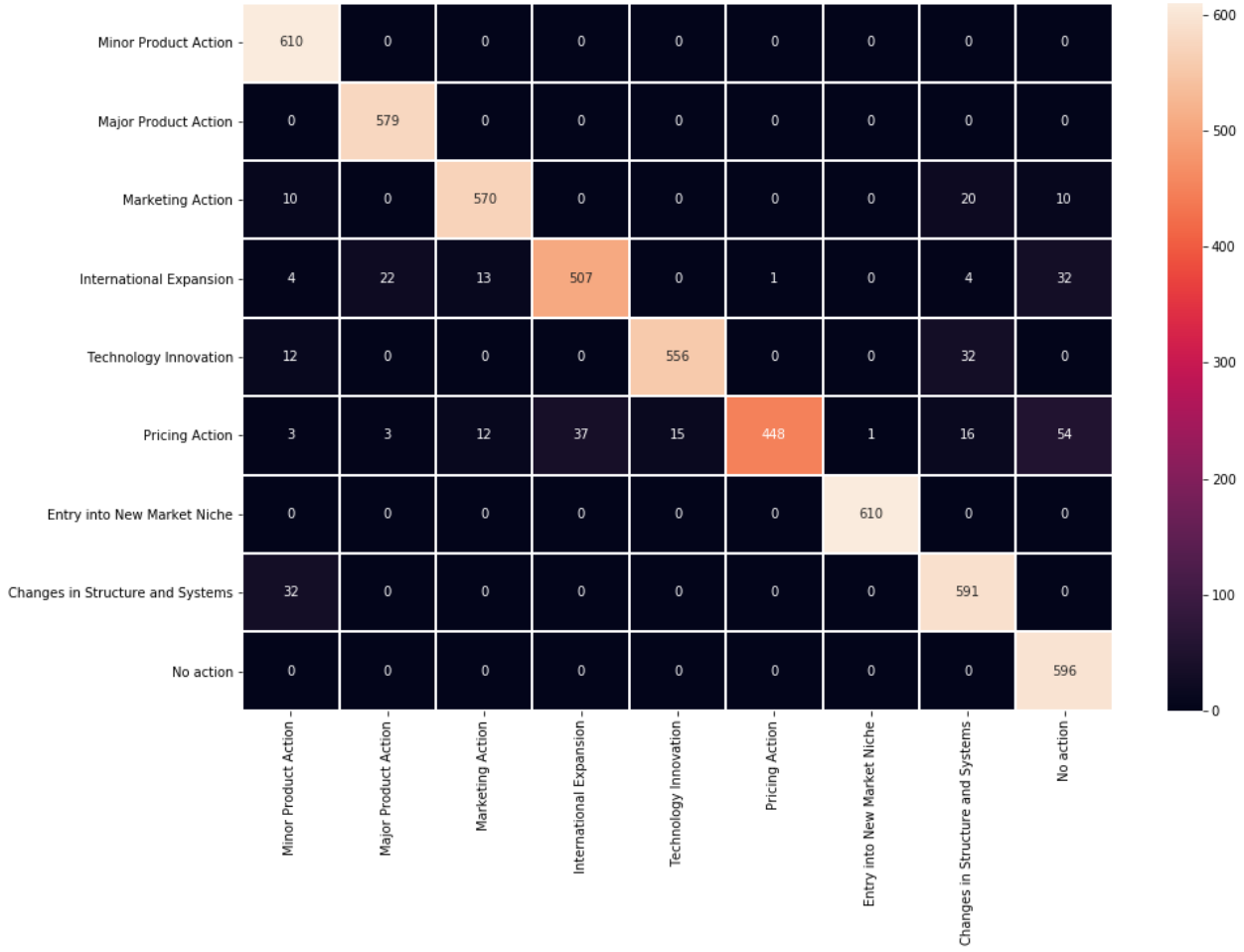

Fig. 6 Confusion Matrix for ANN after removing classes

When looking at the improved models with the dataset that does not have the three noisy classes, “New Product Action”, “Capacity Action” and “Distribution and Services Improvements”, it is seen that the ANN had a 0.29 higher accuracy than the LR model. It also has a 1% increase in Precision where all other measurements are identical to the LR. The final model is therefore still the ANN model. This model is by the research team seen to solve the classification model the best compared to the several other machine learning algorithms.

A general aspect of the data is the amount and quality of the data. The provided data are separated in both text-files and excel sheets. The text-files contained the articles whereas the excel sheet contained the labels for the classified articles. When connecting labels to articles, 1,121 articles had a label. This is in a machine learning context, not seen as a large pool of data. The data augmentation is, therefore, necessary to construct a proper classifier. By using data augmentation, more specifically word shuffling, no new data are added to the dataset and it is therefore only the 1,121 articles with the additional 100 “no action” articles that are shuffled, to make new combinations of the articles containing an identical mix of words. When augmenting all classes to have 2,000 articles, several of the classes are mixed a high number of times, compared to other classes, leading the algorithms to think that the class only contains that specific mix of words. This might lead to poor generalization of the classification ability. While the final algorithm of this research performs well, this might have improved the accuracy and other measurements when testing it on new and unlabeled articles. To solve the problem, more labeled articles are necessary to widen the vocabulary for the algorithms, so they are able to secure a high precision as well as a high recall. The number of articles in each class should also be identical for securing an even weighting of the classes. This problem can be accommodated by special weighting, but ideally, the number of articles in each class should be identical.

Including reflective testing and argumentation, the basic ANN is seen to solve the research objective the best. This algorithm is, therefore, the prominent result of the research combined with its possible uses and future possibilities for improving and optimizing business processes and none the less the classification of news articles into competitive actions solving the major objective of this research.

A. Future Research Objectives

- How to utilize the algorithm on the whole unstructured dataset
- How to optimize the provided data for competitive action classification
- How to integrate the algorithm into a competitive dynamics analysis system

REFERENCES

[1] A. Karnani and B. Wernerfelt, "Multiple point competition," *Strateg. Manag. J.*, vol. 6, no. 1, pp. 87–96, Jan. 1985.
[2] P. Pita Barros, "Multimarket competition in banking, with an example from the Portuguese market," *Int. J. Ind. Organ.*, vol. 17, no. 3, pp. 335–352, Apr. 1999.
[3] T. Yu and A. A. Cannella Jr., "Rivalry between multinational enterprises: An event history approach," *Acad. Manag. J.*, vol. 50, no. 3, pp. 665–686, Jun. 2007.
[4] T. Yu, M. Subramiam, and A. A. Cannella Jr., "Rivalry Deterrence in International Markets : Contingencies Governing the Mutual Forbearance Hypothesis," vol. 52, no. 1, pp. 127–147, 2009.
[5] T. Yu, M. Subramaniam, and A. A. Cannella Jr., "Competing globally, allying locally: Alliances between global rivals and host-country factors," *J. Int. Bus. Stud.*, vol. 44, no. 2, pp. 117–137, 2013.
[6] T. Yu and A. A. Cannella Jr., "A Comprehensive Review of Multimarket Competition Research," *J. Manage.*, vol. 39, no. 1, pp. 76–109, 2012.
[7] M.-J. Chen and I. C. MacMillan, "Nonresponse and Delayed Response to Competitive Moves: The Roles of Competitor Dependence and Action Irreversibility," *Acad. Manag. J.*, vol. 35, no. 3, pp. 539–570, Aug. 1992.
[8] J. W. Pennebaker, M. R. Mehl, and K. G. Niederhoffer, "Psychological Aspects of Natural Language Use: Our Words, Our Selves," *Annu. Rev. Psychol.*, vol. 54, no. 1, pp. 547–577, 2003.
[9] S. R. Das and M. Y. Chen, "Yahoo! for amazon: Sentiment extraction from small talk on the Web," *Manage. Sci.*, vol. 53, no. 9, pp. 1375–1388, 2007.
[10] R. Belderbos, M. Grabowska, B. Leten, S. Kelchtermans, and N. Ugur, "On the Use of Computer-Aided Text Analysis in International Business Research," *Glob. Strateg. J.*, vol. 7, no. 3, pp. 312–331, 2017.
[11] I. Bose and R. K. Mahapatra, "Business data mining - A machine learning perspective," *Inf. Manag.*, vol. 39, no. 3, pp. 211–225, 2001.
[12] C. Coulombe, "Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs."
[13] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, 2017.