Automatic Correspondence Distribution for a Public Institution

André Miguel Balau Fazendeiro
andre.fazendeiro@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

January 2021

Abstract

Correspondence distribution is of utter importance in a large organization such as the Portuguese Navy. A misdirected document might have severe repercussions such as important tasks not being performed and information being lost.

Over time, text classification went from relying solely on word frequency models to sequential models with word embeddings. This paradigm shift is currently the state of the art and reveals promising results in large scale datasets.

Currently, correspondence within the Navy is classified by hand which can be prone to human error and time-consuming. Hence, this dissertation addresses this problem, studying viable alternatives for automatic text classification, relying on Machine Learning and Natural Language Processing tools.

With this goal in mind, various machine learning models were tested and studied, with some of them showing positive results, such as Logistic Regression, with over 90% average accuracy over all labels and an average Exact Match Ratio of approximately 50%.

Keywords: Correspondence Distribution, Multi-label Classification, Natural Language Processing, Machine Learning

1. Introduction

In a world with an ever growing data flux, large organizations demand fast and efficient information management. The lack of an adequate system to maintain and manage this data flow might result in substantial losses for a company's efficiency and productivity. For this reason, computer systems are designed and used to provide the power and accessibility of fast indexation and document security.

The Portuguese Navy currently employs data management strategies performed by humans in a somewhat tedious and outdated manner that increases the risk for human error and subjectivity underlying document classification which may lead to lost documents and difficult retrievals.

This paper presents an alternative to this practice, proposing an automatic correspondence distribution system, capable of classifying documents based on its contents.

The remaining structure of this paper is the following. Sections 2 and 3 provides a brief introduction to the underlying problems for this work, namely Multi-label Classification and Text Classification. Section 4 expands on other articles related to the case under study. Section 5 describes the process of building and testing a machine learning model tailored for multi-label text classification. Section 6 does a brief analysis on the results achieved by the implemented models. Finally, Section 7 summarizes the work done and Section 8 presents ideas for future work.

2. Multi-Label Classification

Multi-Label Classification is the task of selecting from within a list of possible labels, the ones which should be associated with the object we are trying to classify. Multi-Label Classification has multiple use cases, like text and sound categorization [3, 6], semantic scene classification [5], medical diagnosis [30, 9] or gene and protein function classification [8].

Multi-Label Classification differs from classification tasks like Binary Classification and Multi-Class Classification, as the first does not take into consideration dependencies between distinct labels and in the latter, it is only attributed one label per document [20], which does not accomplish our goal.

The challenges of this task are deeply linked with data sparsity and scalability, due to the often high dimensionality of the data, the imbalance, and the dependencies between labels that must be taken into consideration [11].

3. Text Classification

Text classification corresponds to the task of assigning the correct label or labels to a certain text sample. Examples of this task include language identification [7], genre classification [19], sentiment analysis [18, 2] and Spam detection [12]. In the problem described by this dissertation, the aim is to find the correct labels to be assigned to each text document in the dataset.

Working with textual documents requires converting our texts to trainable data. This process is called vectorization and there are multiple ways of achieving it. Two common approaches to representing documents in vector form are bag-of-words (n-grams) and word vectors.

Bag of Words techniques, also known as N-gram models are used to convert documents into vectors. This works by assigning each token a position in the vector and filling that position with the count for the number of times the token appears in the document. In the end, we have for each document a vector containing the number of times each known token appears in that specific document. Using this model we ignore word context and sequence in the document, reducing computing time at the expense of some information loss.

Alternatively, instead of having a vector with frequencies, it is also common to attribute values to each term in the vocabulary. For example, when using tf-idf the weight of each term is calculated and then used instead of flat count frequencies. The tf-idf algorithm is a product between term frequency(tf) – the frequency a word *t* appears in a document *d* – and the inverse document frequency (idf) – the fraction of documents in which the word *t* appears [16]. This is a better alternative to using a count vectorizer being used through this work's experiments.

The disadvantage of N-gram models is not taking into account order in the sequence, which results in information loss. To prevent this loss in text classification it is common to use sequential models. For these models, each token is represented as a vector and the document to represent is nothing more than a sequence of vectors.

How the tokens are vectorized presents the big distinction for these models. On the one hand, we have the classic approach using One-Hot Encoding - attributing one position in the vector for each token in the vocabulary and filling it with a value corresponding to that token's occurrence.

On the other hand, we can make use of Word Embeddings. This method, which recently has been widely used consists in giving each word a dense vector representation, representative of its position in semantic space. Such embedding weights can be trained in the dataset from scratch, be imported from another corpus or be pre-trained in one corpus and then fine-tuned for the dataset in question through transfer learning.

4. Related Work

Working on the problem of Multi-Label Classification, it is common to divide the approaches in two categories: Problem Transformation and Algorithm Adaptation. The former transforms the problem so that it can be solved by traditional classification algorithms while the latter makes use of such algorithms which are adapted to do multi-label learning. The main reasons for choosing Problem Transformation is the ease to test and to classify using many common used algorithms. Otherwise, we might be more interested in using Algorithm Adaptation approaches, considering these are often more suited to the problem and take into consideration things like label correlation, which tends to be overlooked in some Problem Transformation approaches [11].

Apart from Problem Transformation and Algorithm Adaptation, some authors [23] still hold a special category for solutions that make use of ensemble methods to predict labels.

4.0.1 Problem Transformation

In the category of Problem Transformation, some solutions while being conceptually simple to implement still reveal promising results. One example of such approaches is Binary Relevance [35, 32] which simply consists in training one binary classifier for each label. The advantages are the conceptual simplicity, not being constrained to a particular learning technique, so almost every single-label classifier can be used as the underlying model, with models based in Support Vector Machines [17, 28] and Naive Bayes [28], being successful. Besides that, they are able to learn from partially labeled instances, since each classifier is trained independently. The major drawback is the scalability, considering the number of classifiers grows linearly with the number of labels, being unusable in very large datasets (Extreme Multi-Label Classification) [35]. Another issue appointed to Binary Relevance is not taking into consideration label correlations, which are important for multi-label classification [35].

There are some algorithms that base themselves in Binary Relevance and try to account for the correlation part. One of these methods is Classifier Chains [27] which similarly to Binary Relevance trains one classifier per label, however these classifiers are trained sequentially and take as input the instance to classify plus all the labels resultant of the previous classifiers. As mentioned, this has the advantage of accounting for the correlation while maintaining a computing complexity close to Binary Relevance, with the disadvantage of not being possible to do parallel training to reduce computing time, or train in incomplete data [27].

Other Problem Transformation methods include Label Powerset methods [32], which transform the multi-label problem into a single-label one by training one classifier for each possible combination of labels. The advantage of this method is that it takes into consideration correlation between labels, but deeply suffers from the problem of data scarcity, since some label combinations might not have enough representation in the dataset and the results might not be distinguishable from Binary Relevance, despite being expected to show better results for taking into account label correlation [32].

For Algorithm Adaptation approaches, there are techniques altered in a way to directly solve the Multi-label classification problem, and are generally better at taking things such as correlation between labels into consideration. Some Algorithm Adaptation may be bound to a particular learning method, however, they can be as simple as using a Problem Transformation method internally or collecting multiple classification confidences and joining them [27]. For such approaches we can see ones using traditional learning methods as well as recent deep learning approaches.

For classical classification methods, one can find adaptations for Decision Trees, Support Vector Machines and Instance Based Classifiers [11]. An example of the last method is an adaptation of k-Nearest Neighbors to the Multi-label classification [34] problem, a lazy learning method that shows very promising results in several datasets.

More recent methods using Artificial Neural Networks often focus on the scalability and data sparsity often associated with the Multi-Label learning problem, often referred to as Extreme Multi-Label Learning problems [14, 20]. Using such methods we see very different approaches, focusing on the loss functions resulting in a Precision@5[1] of 48.08% in the *EUR-Lex*[2] dataset [14], using different types of Artificial Neural Networks, such as applying word embeddings followed by a Convolution Neural Network resulting in a Precision@5 of 51.41% in the same dataset [20] or using restricted Boltzmann machines that help improve the feature-space [25], this one not focused at Extreme Multi-Label, resulting in an accuracy of 0.742 against 0.770 compared with ECC in the *Medical* Dataset, 48.0% to 45.4% on *Enron* Dataset and 45.1% to 46.1% on *Reuters*, all these text datasets [25].

4.0.2 Ensembles

Other examples that are often included in a group of its own and are known for increasing overall accuracy, overcoming over-fitting and allowing parallelism are Ensemble Techniques [27].

One such example of this method is the Ensemble of Pruned Sets [26]. For this method, we begin by building pruned sets, which consist of only the most relevant label relationships in the training set. After pruning the least frequent sets, we are left with the label groups that have a significant representation in the set. After that, a binary classifier is trained for each set, similar to what happens with Label Powerset. The additional advantage of including an Ensemble is reducing overfitting and the possibility of assigning sets of labels that were nonexistent in the training data [26].

The Ensemble of Classifier Chains [27], as the name points out, is a grouping of Classifier Chains, and in training, each chain is assigned a random chain ordering and a random subset of the dataset [27].

One more Ensemble method is RAkEL [33], or *RAndom k-labELsets*, which in its turn, is a grouping of Label Powerset classifiers. For this method, we have two changeable parameters, k and m, which represent the length of the label sets to be considered and the number of iterations, respectively.

5. A Classifier System for Correspondence Distribution

5.1. Dataset Analysis

The provided dataset contains 7300 documents, from which 7185 possess a distribution table. The distribution table is the target we want to predict and contains the following seven labels, corresponding Portuguese Navy's units:

- STI - Superintendência das Tecnologias da Informação (IT Superintendency)
- CDIACM - Centro de Documentação de Informação e Arquivo Central da Marinha (Navy's Central Information and Archive Documentation Center)
- DAGI - Direção Análise e Gestão de Informação (Direction of Analysis and Information Management)
- DITIC - Direção de Tecnologias de Informação e Comunicações (Information and Communication Technologies Management)
- C/GAB - Gabinete do Superintendente das TI (IT Superintendent's Office)

[1] metric mostly used in Information Retrieval problems, measures the ratio of relevant results in the top five predictions – in this case the five labels with higher confidence

[2] *EUR-Lex* Dataset with 19348 instances, 3993 labels and a label cardinality of 5.31 [14]

- SERV.PART - Serviço Particular (Private Service)
- ADJ.SEC1 - Entidade Contabilística (Accounting Entity)

For each of the listed departments, a character is assigned as a descriptor to the degree of action required from that specific department:

- Blank (0) - if no action is required and that department is not a recipient
- Letter C - if the document in question is for information (*Conhecimento*) but no action is required from said department
- Letter A - if the document is to be addressed to said department and the latter is required to take action (*Ação*)

Documents can be of two different types, depending on their layout: standardized or non-standard. Standardized documents follow a specific layout defined by the Navy, and have their text fields segmented into distinct tags in a XML file. Between these tags we have: Header, Title, Body, Signature, Attachment, etc. From the experiments carried out, using only Title and Body of the document provided the best results. For non-standard documents, all the content is in a single Text tag and these documents correspond to invoices, receipts, faxes, memos, diplomas and emails and represent approximately 35% of the dataset documents [29].

The dataset contains documents from years 2014 through 2019 and by setting aside years 2018 and 2019 we get a test set with 26.34% of the samples and temporal relevance. In Table 1 some key metrics are displayed. It is noticeable that class *A*, the minority class is much less represented than classes *C* and *blank*.

Table 1: Group of key metrics from the Portuguese Navy dataset

Metric	Value
Number of Samples	7185
Number of Labels	7
Number of Classes per Label	3
Average Samples **blank**	4995
Average Samples **C**	1892
Average Samples **A**	298
Median Number of Words per Sample	50

This uneven distribution can be further and better understood by visualizing the following Figure 1, which represents the distribution of classes within each label. From this graph, we can see that label *DITIC* is the only one that deviates from a majority class *blank* and minority class *A*. For this label *C* is the most represented class and classes *blank* and *A* are relatively close, even though class *A* is the least represented in every label, with labels *CDIACM* and *ADJ.SEC1* counting only with twelve and four class *A* samples, respectively.

Table 2: Most common uni and bi-grams, excluding stop-words.

N-gram	Frequency
marinha	13141
informação	9908
nacional	9644
contrato	9513
data	9004
lisboa	8928
defesa	8539
serviços	8412
pt	8232
ser	6857
chefe	6505
artigo	6475
defesa nacional	6350
gestão	6315
total	6262
valor	6098
serviço	5879
comunicações	5698
formação	5681
tecnologias	5626

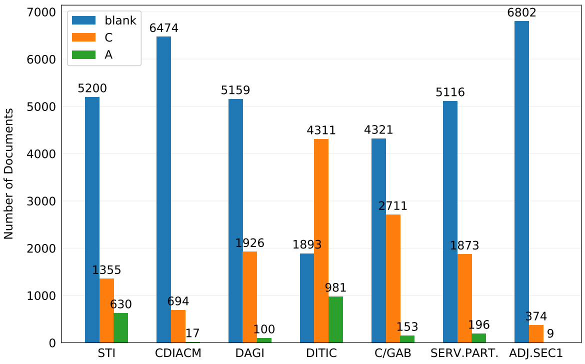

Figure 1: Bar plot with the distribution of classes for each department in the dataset.

In Table 2 we can visualize the twenty most common words found in the documents, after removing stop-words and lowering the case. By analyzing these words we can notice the most used vocabulary is quite specific to the context of the Navy and managing national and international affairs.

From looking into Figure 1 and 1 it is predictable that the biggest challenge to having a robust classifier is the dataset imbalance, most prominent in class *A*.

5.2. Model Selection

Looking into the approaches discussed in related work and transferring them to our challenge, two distinct ways to tackle the problem rise.

The first suggested approach consists in splitting each individual department degree (Blank, A and C) into separate labels, while accounting for the exception that each department can only have one degree and then treating it just like a Multi-Label Classification problem. The other approach, and the one more exhaustively pursued throughout this dissertation was to treat the classification as in the Binary Relevance method, considering each label an independent multi-class classification problem, selecting for each department one degree, either *A*, *C* or *blank*.

Several classifiers were implemented, in order to evaluate their performance in our dataset. The implemented models are the following:

- Baseline Classifier - Most Frequent Tag: A dummy estimator that classified every instance as the majority class.
- Logistic Regression: Fits the data through a linear regression model and then computes the probability of our sample belonging to each class. Logistic Regression has displayed great performance in text classification tasks, especially when compared with other classic methods [24].
- Naive Bayes (NB): Calculates the probability of a given sample belonging to each particular class in a simplistic way, by taking into consideration the prior probabilities for each class and the conditional probabilities of seeing the input, given each class. Naive Bayes is often used for spam filtering [13].
- Support Vector Machines (SVMs): Finds the linear hyper-plane that best discriminates between classes
- Decision Trees and Random Forests: Tree based models work by partitioning data into cuboid regions. Decision Trees have the tendency to overfit and so an ensemble of Decision Trees (a Random Forest) is ofter used instead. This is an ensemble method that trains numerous Decision Trees in subsets of our data.
- k-Nearest Neighbors: Using a distance metric, finds the closest samples to the unseen instance in the dataset and classifies that instance by doin a majority vote between the closest *k* neighbors.
- Multi-Layer Perceptron: an Artificial Neural Network, composed of multiple layers of Perceptrons that work in a Feedforward Network.

After watching promising results using these models with a bag-of-words approach, the decision to try sequential models was made, given their recent success in the field [22, 10]. For this reason we tested with embeddings and a Separable Convolutional Neural Network (SepCNN). The experiments made included embeddings trained from scratch, and using FastText's pre-trained word embeddings for Portuguese [15, 4], trained on Wikipedia data[3]. from the FastText corpus and loaded from FastText and fine-tuned on our dataset.

SepCNN is a neural network specialized for text classification tasks that works with word embeddings and uses one-dimensional depthwise separable 1D convolutional layers. Depthwise separable convolutions reduce the computation time needed for the convolutions and the number of parameters to tune. They work by dividing the traditional convolution into a depthwise convolution (filtering step) followed by a pointwise convolution (combination step).

6. Results

6.1. Methodology

All models were compared using the same methodology. Initially a random split was made for a train and test static sets that was the same for each of the experimented models, for convenience and in order to have a reference performance for further evaluations. Subsequently, a test set was detached from the remaining documents which contained the most recent documents of the dataset (those belonging to the year 2018 and 2019). The remaining elements of the dataset were divided into 4 splits to be used for cross validation.

Several evaluation metrics were used in order to better evaluate the results. The most referred to were accuracy, exact match ratio and recall.

The most referred metrics were the following:

- Exact Match Ratio (EMR)- Measures how many instances were completely well classified. Ignores partially correct instances [31].

$$\mathrm{EMR} = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i)$$

- Accuracy - Measures the proportion of correct labels [31].

$$\mathrm{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

- Recall - Ratio between the predicted correct labels and all true labels, averaged over all instances. A low Recall value in minority classes might reveal a bias towards the majority classes [31].

[3] https://dumps.wikimedia.org

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{Yi}$$

6.2. Discussion

6.2.1 Bag of Words Representation Approach

The results for the discussed classifiers can be summarized in Table 3. The best performing classifier is Logistic Regression, followed closely by Support Vector Machines and Random Forests. On the other end of the spectrum, not considering the baseline, are Naive Bayes and Decision Trees. Naive Bayes is often used as a baseline method, given its simplicity in implementation. The tendency of Decision Trees has also been appointed and is a likely reason for this classifier's weak scores, which had a major improvement when the Decision Trees ensemble method, Random Forests, was tested.

6.2.2 Sequential Representation Approach

The sequential models trained were also among the worst performing classifiers. SepCNN with FastText vectors out of the box was the worst of the three sequential classifiers. This shows that FastText vectors, despite being pre-trained in a much bigger corpus were still beaten by word vectors fitted strictly in the training set. However, when pretraining the same FastText word vectors the results exceeded training from scratch, showing an advantage in training in a larger corpus, with an average accuracy increase of approximately 4.4% over FastText raw vectors and 0.5% average accuracy increase over trained from scratch word vectors.

When working with sequential models, using word vectors led to results below average, despite the rising popularity of this technique. Reasons for this seem to be linked to the specificity of the vocabulary, being most of it related to Navy's themes and the objective concise writing removes most of the ambiguities that would make word vectors trained in a large Corpus overcome other methods. Word vectors seem to work best when presented with a wider vocabulary and ambiguous texts, where a context is important to extract a meaning.

Another factor is the ratio between the number of samples and number of words per sample, which according to this source should be taken into consideration. When this ratio is higher than 1500, SepCNN seems to outperform Multi-Layer Perceptron and other classic methods. Below that value, Multi-Layer Perceptron performs better [1]. Taking our dataset statistics, presented in Table 1, into consideration, this ratio is of approximately 145, roughly 10 times less than the suggested value, indicating a shortage of samples to train an appropriate sequential model.

6.3. Dataset Imbalance

One of the biggest challenges for the developed methods was fighting the dataset imbalance. When plotting the confusion matrix for the best performing classifier (Logistic Regression) our attention is drawn to the lesser recall for class *A*, where a tendency towards predicting classes *C* or *blank* is verified approximately one third of the times the true class is *A*.

In order to fight this imbalance, multiple sampling methods were tested, in order to analyse how the results would be affected. The following methods were implemented:

- Imbalanced Set: This is the control test, corresponding to the unchanged dataset.
- Undersampling: Aims at balancing sets by ignoring certain samples. May lead to huge information loss.
- Oversampling: Resamples the minority classes, giving every class the same representation in the dataset.
- EasyEnsemble [21]: Creates multiple subsets where the minority class is reused in each subset and the majority classes undersampled.

In Table 4 we can see the results for each one of the data balancing methods. As can be witnessed in Table 4 none of the methods performed better than the imbalanced set, however Oversampling provided really close results. This means it can be considered as a viable option to creating a balanced training in an imbalanced set so that every class has the same weight in the decision. This fact is more evident when looking into the confusion matrices present in Figure 2, as an average between labels. It is visible that Oversampling is a good balancing method if the purpose is to increase recall and the number of positive classifications for the minority classes, at the expense of some accuracy.

Another thing to notice is the considerably lower performance of the Undersampling approach, reason for this is the residual number of samples for each class in some cases and the huge information loss resulting from not taking into consideration every available sample.

6.4. Final Evaluation

Since we wanted to avoid creating a bias towards any of the classifiers, we isolated a test set, comprised of the samples from the years 2018 and 2019. Training then our top classifier in the whole

Table 3: Results for the classifiers trained in the Portuguese Navy dataset (cross-validated).

	Baseline	NB	LogReg	MLP	SVM	Rforests	Decision Tree	kNN	SepCNN	Sep FT	Sep FT Train
EMR	15.56%	22.09%	**49.23%**	46.86%	48.00%	46.73%	36.95%	37.43%	30.85%	23.26%	32.68%
Accuracy	74.52%	80.16%	**90.08%**	89.41%	89.59%	89.22%	86.40%	86.20%	84.50%	80.61%	85.07%

Table 4: Results for Data Balancing methods from a random sample.

	Logistic Regression			
	Imbalanced Set	UnderSample	OverSample	EasyEnsemble
STI	90.71%	86.55%	**90.92%**	87.76%
CDIACM	**96.67%**	70.35%	96.04%	76.97%
DAGI	91.09%	72.72%	**91.21%**	74.89%
DITIC	**85.38%**	78.76%	84.34%	80.72%
CGAB	**89.55%**	75.43%	89.13%	80.34%
SERV PART	**85.71%**	66.56%	85.21%	75.93%
ADJ SEC	**95.96%**	19.58%	94.79%	45.86%
Average Accuracy	**90.72%**	67.13%	90.24%	74.64%
EMR	**59.64%**	4.37%	58.31%	21.57%

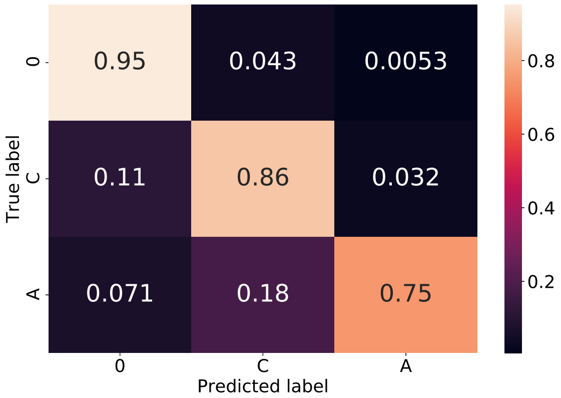

Figure 2: Normalized Confusion Matrix for the best performing classifier averaged over every label on an oversampled set.

training set, using the same parameters as in development we obtained an average accuracy of 82.02% and an Exact Match Ratio score of 35.40%. This is quite a decrease when compared to previous tests. However, such decline in performance can be associated to the temporal nature of the data, since through all the years in the dataset, writing methodologies and overall data handling principles have changed. This demonstrates the importance of keeping the dataset updated, in order to achieve optimal results.

6.5. Summary

Regarding the models implemented, we can see from the results in Table 3 that the classifier displaying best results was Logistic Regression, followed closely by the Multi-Layer Perceptron and also Support Vector Machines.

The lowest scores between classifiers (not taking into consideration the Baseline) were observed in both Naive Bayes – for which said results can be justified by the simplicity of the classifier and its assumptions – and the sequential models, which suffer from the depth of the vocabulary against its breadth, consisting in documents within the same context and with low ambiguity and the fact that the designed models could take advantage of more samples to train on.

Looking into the dataset imbalance it might be valuable to use a balancing method such as oversampling. While this mechanism displayed slightly worse results than for the imbalanced set, it is a good alternative for a system in which we are looking for higher recall for the minority classes, especially given the fact that such classes are the source of biggest confusion for our classifiers.

7. Conclusions

This dissertation explored the correspondence distribution panorama taking place currently in the Portuguese Navy. This process is done by hand, prone to human error and subjectivity, along with the extra time consumption required to perform the task. These effects are undesirable and can hinder the productivity of a large organization such as the Portuguese Navy.

This work tested several classifiers in multiple settings, in order to select the best to perform this task. Analysis of the results revealed the major liabilities linked to this task, such as the imbalance between classes, with one of them being in much minor representation, and the temporal factor of the data which was demonstrated by the decrease in performance, when evaluating results in a set separated by belonging years.

This work explored the influence of document representation and sampling methods to find the classifier that outperformed the others. With this goal in mind, this dissertation presents a classification method for automatic correspondence distribution using Logistic Regression and the principles of Binary Relevance for Multi-label Classification with the aid N-gram models that displayed an average accuracy of 82.02% over seven distinct labels, correctly predicting all seven labels for 35.40%. Regarding the imbalance of the dataset, an alternative using Oversampling was also presented, displaying little accuracy decrease when compared to the imbalanced set and a significant increase in recall.

8. Future Work

For future work, there are various ideas that could help provide better results for this task and improve the classification performance. For starters, there is some noise being introduced through the OCR, producing illegible words that affect the quality of the dataset.

After that, regarding data pre-processing, it would be a good idea to try Stemming to give more weight to context and less to specific words and to implement Named Entity Recognition, since when looking into the features with more weight to classification, some classifiers presented personal names. It would be a good idea to replace said names by the person's rank within the Navy in order to prevent biases towards rotating roles. Besides that, it would be good to have more documents to train on, in order to fully experiment with more complex models which for this task were not the adequate choice.

References

[1] Step 2.5: Choose a model nbsp;—nbsp; ml universal guides nbsp;—nbsp; google developers, Oct 2018.

[2] M. F. R. A. Bakar, N. Idris, L. Shuib, and N. Khamis. Sentiment analysis of noisy malay text: State of art, challenges and future work. *IEEE Access*, 8:24687–24696, 2020.

[3] M. M. Bittencourt, R. M. Silva, and T. A. Almeida. Ml-mdltext: A multilabel text categorization technique with incremental learning. In *8th Brazilian Conference on Intelligent Systems, BRACIS 2019, Salvador, Brazil, October 15-18, 2019*, pages 580–585. IEEE, 2019.

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[5] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognit.*, 37(9):1757–1771, 2004.

[6] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen. Polyphonic sound event detection using multi label deep neural networks. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, pages 1–7. IEEE, 2015.

[7] D. W. Castro, E. Souza, D. Vitório, D. Santos, and A. L. Oliveira. Smoothed n-gram based models for tweet language identification: A case study of the brazilian and european portuguese national varieties. *Applied Soft Computing*, 61:1160 – 1172, 2017.

[8] A. Chan and A. A. Freitas. A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In M. Cattolico, editor, *Genetic and Evolutionary Computation Conference, GECCO 2006, Proceedings, Seattle, Washington, USA, July 8-12, 2006*, pages 27–34. ACM, 2006.

[9] H. Chougrad, H. Zouaki, and O. Alheyane. Multi-label transfer learning for the early diagnosis of breast cancer. *Neurocomputing*, 392:168 – 180, 2020.

[10] F. Dernoncourt. *Sequential short-text classification with neural networks*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA, 2017.

[11] E. Gibaja and S. Ventura. A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3), 2015.

[12] T. S. Guzella and W. M. Caminhas. A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 36(7):10206 – 10222, 2009.

[13] J. Hovold. Naive bayes spam filtering using word-position-based attributes and length-sensitive classification thresholds. In S. Werner, editor, *Proceedings of the 15th Nordic Conference of Computational Linguistics, NODALIDA 2005, Joensuu, Finland, May 2005*, pages 78–87. University of Joensuu, Finland, 2005.

[14] H. Jain, Y. Prabhu, and M. Varma. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu:935–944, 2016.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[16] D. Jurafsky and J. H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.

[17] J. Li, F. Alzami, Y. Gong, and Z. Yu. A multi-label learning method using affinity propagation and support vector machine. *IEEE Access*, 5:2955–2966, 2017.

[18] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14 – 23, 2014.

[19] C. S. Lim, K. J. Lee, and G. C. Kim. Multiple sets of features for automatic genre classification of web documents. *Information Processing Management*, 41(5):1263 – 1276, 2005.

[20] J. Liu, W. C. Chang, Y. Wu, and Y. Yang. Deep learning for extreme multi-label text classification. *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124, 2017.

[21] X. Liu, J. Wu, and Z. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B*, 39(2):539–550, 2009.

[22] A. Madasu and V. A. Rao. Sequential learning of convolutional features for effective text classification. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5657–5666. Association for Computational Linguistics, 2019.

[23] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084–3104, 2012.

[24] T. Pranckevičius and V. Marcinkevičius. Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification. *Baltic Journal of Modern Computing*, 5(2):221–232, 2017.

[25] J. Read and F. Pérez-Cruz. Deep learning for multi-label classification. *CoRR*, abs/1502.05988, 2015.

[26] J. Read, B. Pfahringer, and G. Holmes. Multi-label classification using ensembles of pruned sets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 995–1000. IEEE Computer Society, 2008.

[27] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Mach. Learn.*, 85(3):333–359, 2011.

[28] S. M. Rendón, D. H. Peluffo-Ordóñez, and G. Castellanos-Domínguez. support vector machine-based aproach for multi-labelers problems. In *21st European Symposium on Artificial Neural Networks, ESANN 2013, Bruges, Belgium, April 24-26, 2013*, 2013.

[29] G. A. Rodrigo. Projeto: Identificação e Classificação de Entidades Mencionadas e Eventos em Documentos da Marinha. 2020. Master Thesis. IST, UL.

[30] H. Shao, G. Li, G. Liu, and Y. Wang. Symptom selection for multi-label data of inquiry diagnosis in traditional chinese medicine. *Sci. China Inf. Sci.*, 56(5):1–13, 2013.

[31] M. Sorower. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, pages 1–25, 2010.

[32] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee. A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science*, 292:135–151, 2013.

[33] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4701 LNAI, pages 406–417, 2007.

[34] M. Zhang and Z. Zhou. A k-nearest neighbor based algorithm for multi-label classification. In X. Hu, Q. Liu, A. Skowron, T. Y. Lin, R. R. Yager, and B. Zhang, editors, *2005 IEEE International Conference on Granular Computing, Beijing, China, July 25-27, 2005*, pages 718–721. IEEE, 2005.

[35] M. L. Zhang, Y. K. Li, X. Y. Liu, and X. Geng. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12(2):191–202, 2018.