Application of the Multi-label Residual Convolutional Neural Network text classifier using Content-Based Routing process

Safa ELKEFI, MSc\textsuperscript{1,3,*} and Achraf TOUNSI, MSc\textsuperscript{2,3,4}

\textsuperscript{1} Industrial Engineer, National Engineering School of Tunis, Tunisia
\textsuperscript{2} Modelling for Industries and Services (MIndS), National Engineering School of Tunis, Tunisia
\textsuperscript{3} Stevens Institute of Technology, Hoboken, NJ
\textsuperscript{4} Infolegale & Marketing, Lyon, France
\* Corresponding author : selkefi@stevens.edu

Abstract—In this article, we will present an NLP application in text classifying process using the content-based router. The ultimate goal throughout this article is to predict the event described by a legal ad from the plain text of the ad. This problem is purely a supervised problem that will involve the use of NLP techniques and conventional modeling methodologies through the use of the Multi-label Residual Convolutional Neural Network for text classification. We will explain the approach put in place to solve the problem of classified ads, the difficulties encountered and the experimental results.

Keywords—Neural Network, Text Classification, Contet-Based Router, Multi-label, Optimization...

I-INTRODUCTION

The use of neural networks in language processing applications is still an open problem in research, which needs to be addressed in more depth, especially on task-specific issues (which architecture corresponds to which tasks), more global optimizations and learning transfer techniques in order to be able to use networks already trained on new problems easily. One of these subjects is the application of neural language with routing processes. In Infolegale’s production processes, the use of data science methods to process the text of documents in natural language is a necessity. In fact, they automatically extract the exploitable data and therefore meet the needs of the COMPLIFY projects. The project is part of a global intensification of the fight against money laundering, corruption and fraud. The company wishes to accompany its clients in the mapping of their risks, including the knowledge of the client and the identification of the person(s) with the control of the company (beneficial owner). Scanning of legal announcements, prediction of events and extraction of named entities were put in place to answer the first part of the project. The goal is to be able to make the process of acquiring legal announcements faster and faster. The process is nowadays complicated and relies on 724 newspapers to be collected and about 1 200 000 advertisements per year to dematerialize, to codify, to inform in a database and to sirenize then.

II-CONTEXT OF THE PROJECT

The company in question is revolutionizing its acquisition process of daily articles and newspapers by applying state-of-the-art NLP models in order to automate the whole process. In order to achieve such goal, the mission was defined during my end of study project where I worked on setting up a set of models that will determine the events contained in the legal announcements according to the types of events standardized by Infolegale. And to accomplish this, we will use NLP models with Content-Based approach in order to route the events into their corresponding sphere.

III-METHODOLOGY

1 Understanding the target output

Once the legal ads have been extracted from the logs and PDF articles, the second process begins: identifying the events described by the ad. The objective of this part is to set up a set of models that will determine the events contained in the legal announcements according to the types of events standardized by Infolegale. And to accomplish this, we will use NLP models with Content-Based Routing process in order to classify each of the 204 event that could be found in an ad.

2) Convolutional neural network

Convolutional neural networks or CNNs\textsuperscript{6} can be defined as a class of artificial networks of deep-feed artificial
neurons. This is an algorithm with layers stacked so that input information at the beginning can be refined carefully in all layers in a unidirectional order until the desired result is achieved.

CNN architectures uses the layers presented as follows[6]:

- Convolutional layers; used to preserve the spatial orientation of entities in an image.
- Grouping layers; used to downsample an image (reduce it).
- Fully connected layers; to stack the convolution / convolution layer output on a single-column matrix and compress it continuously to obtain a smaller but reasonable output.
- Soft-max output; to present the output generated by the fully connected layers.

In the next section, we define a convolutional neural network subtype which is the residual convolutional neural network.

3) Residual Convolutional Neural Network

A residual convolutional neural network [1] is an artificial neural network (ANN) of a type that relies on known constructs from pyramidal cells of the cerebral cortex. To do this, residual neural networks use skipped connections or shortcuts to jump over certain layers. [3]

![Fig. 1: Canonical a residual neural network form](image)

Figure 1 simplifies the step of passing an L-1 layer of L-2 activation. These typical models are implemented with double or triple settes containing non linearity and batch normalization between the two.

![Fig. 2: Structure of a residual neural network](image)

Figure 2 describes the structure of a network of residual neurons. The main idea behind this network is the residual block.[1] The network allows the development of extremely deep neural networks, which can contain 100 or more layers.

The equations that describe this structure are:

Simple network:

\[
\begin{align*}
a^{[l]} &= g(z^{[l-1]}) \\
z^{[l+1]} &= W^{[l+1]}a^{[l]} + b^{[l+1]} \\
a^{[l+1]} &= g(z^{[l+1]}) \\
z^{[l+2]} &= W^{[l+2]}a^{[l+1]} + b^{[l+2]} \\
a^{[l+2]} &= g(z^{[l+2]})
\end{align*}
\]

(1) \hspace{5cm} (2) \hspace{5cm} (3) \hspace{5cm} (4) \hspace{5cm} (5)

Residual network:

\[
\begin{align*}
a^{[l]} &= g(z^{[l-1]}) \\
z^{[l+1]} &= W^{[l+1]}a^{[l]} + b^{[l+1]} \\
a^{[l+1]} &= g(z^{[l+1]}) \\
z^{[l+2]} &= W^{[l+2]}a^{[l+1]} + b^{[l+2]} \\
a^{[l+2]} &= g(z^{[l+2]} + a^{[l]})
\end{align*}
\]

(6) \hspace{5cm} (7) \hspace{5cm} (8) \hspace{5cm} (9) \hspace{5cm} (10)

Where \(a^{[l]}\) is the activations or outputs of neurons in layer \(l\), \(g\) the activation function for layer \(l\) and \(W^{[l+1]}\) the weight matrix for neurons for the layer \(l - 1\).

Absent an explicit matrix \(W^{[l+2]}\) (aka ResNets), forward propagation through the activation function simplifies to the equation (10).

With this extra connection, gradients can travel more easily backwards. [3] It becomes a flexible block that can increase the capacity of the network or simply turn into an identity function that does not affect the formation.[3]

4) Content-Based Routing

The end of the task is to set up a system for order processing. We first confirm the order when an incoming order is issued and then check that the ordered item is available in the warehouse. The stock system performs this task. This processing stage series is a perfect candidate for the design of Pipes and Filters.[7] We create two filters, one for the validation phase and one for the inventory system, and route through both filters the incoming messages[7]. Nevertheless, there are more than one distribution system in many business integration scenarios where each system can manage only specific items[7].

![Fig. 3: Example of an inventory check and order routing process](image)

The Content-Based Router[7] checks the content of the message and routes the message to another channel based on the message’s information. The routing can be based on a number of factors, such as field presence, specific field values, etc. Great care must be taken when installing a content-based router to make it easy to control the routing function.
as the router may become a point of regular maintenance. The Content-Based Router can take the form of a configurable rules engine that determines the destination channel based on a set of configurable rules in more complex integration scenarios [7].

5) Input Data Analysis

The data that will be used throughout our study will be a set of random legal ads taken from the local database of the company. The data is consists of raw french text with infinity word possibilities and no clear pattern of string format. The ad contains one or more event type which are needed to be detected. At Infolagale, there are 161 types of events in 7 family types. The volume and variety of legal notice events has prompted us to put in place a decision tree through which an ad will have to go to determine the events in question. In order to implement the decision tree, we will be using a the content-based routing process as described in a previous paragraph.

During this module, we faced many difficulties, namely:

- Database problems: the existence of duplicates of ads with a different identifier, the pollution of results with input errors...
- The analysis of false positives: this step asked us the expertise of the quality team. Not available, it was difficult to find a niche to perform the analysis.
- Grouping events: this step had us take a long time to set up the decision tree. Indeed, grouping ads is not easy and requires a lot of business expertise and performance testing for each attempt.

6) Multi-label Residual Convolutional Neural Network text classifier using Content-Based Routing implementation process

Events grouping

In order to set up our decision tree, the different event types contained in the ads have been analyzed in order to re-group them by the context meaning. Even though the data presented a huge variety in the content, a primary decision tree was put in place. Each of the 7 event families regrouped one or more sub-levels of groups and sub-groups and that is to assure the best model classification process and to predict the corresponding type of event with the highest accuracy, precision and recall possible. The tree is consisted of 43 node with 5 levels of precision and 161 events grouped by their meaning and text content. [7] The tree was then implemented using nested dictionaries and sub-classes with Python. Each node is consisted of a group name, a cutoff, a group description and most importantly, the children. In a matter of fact, the group children will be responsible for redirecting the original ad to the corresponding event. Due to the critical aspect of this point of the project, a high cutoff value is selected to each event (more than or equal to 0.999).

Model Building

In order to optimize our Multi-label Residual Convolutional Neural Network, we will be using the Stochastic gradient descent (SGD) in order to minimize the sum of squared residuals [9]. The algorithm is very useful in cases where the optimal points cannot be found by equating the slope of the function to 0.

One other point, the data set that we will use contains different sizes of events and subgroups, which means that our deep learning neural network model is more likely to quickly over-fit a training data set with few examples, which results in poor performance when the model is evaluated on new data. [2] To overcome this problem, the dropout has the effect of making the schooling system noisy, forcing nodes within a layer to probabilistically take on greater or less responsibility for the inputs. This conceptualization suggests that perhaps dropout breaks-up conditions where community layers co-adapt to correct mistakes from prior layers, in turn making the model extra robust. [2]

Dropout simulates a sparse activation from a given layer, which interestingly, in turn, encourages the community to absolutely research a sparse representation as a side-effect. [4] As such, it is able to be used as an opportunity to pastime regularization for encouraging sparse representations in auto encoder models. [1] The hyper-parameter is introduced which specifies the probability at which outputs of the layer are dropped out, or inversely, the probability at which outputs of the layer are retained. We will give the probability of 0.25 to our dropout value for retaining the output of each node in a hidden layer.

Model Training

A set of 2.3 Million line of ads has been used to train the model, regrouping most of the ad types. The groups and sub-groups varies in size, ranging from 30 to 400 000. The presented model is designed to be effective in this particular case. It will shuffle our data set, divide it on multiple pre-sized batches and iterates over the whole data sets. The size configuration of the batches are taken from 512 to 1024 using a 1.001 compounding factor. once the model is trained over all batches, it will retrain itself and will iterate the same process 100 times in order for the losses to converge to the minimum. The values are selected in a way to optimize the usage of our
GPU machine with a maximum of efficiency[8].

7) Experimental Results and Performance Analysis

After the first model training and implementation, the results were quite impressive and high for the 3 metrics (accuracy, precision and recall). The first results per family are as follows:

| Cutoff | 0.9 | 0.99 | 0.999 |
|--------|-----|------|-------|
| event_code | precision | recall | precision | recall | precision | recall | Support |
| '1100' | 0.745 | 0.985 | 0.975 | 0.918 | 0.780 | 0.707 | 1661 |
| '2xxx' | 0.356 | 0.998 | 0.561 | 0.986 | 0.569 | 0.891 | 19502 |
| '3xxx' | 0.474 | 0.995 | 0.502 | 0.988 | 0.546 | 0.963 | 5552 |
| '4xxx' | 0.304 | 0.992 | 0.314 | 0.976 | 0.324 | 0.913 | 1716 |
| '5xxx' | 0.775 | 0.999 | 0.769 | 0.999 | 0.77 | 0.992 | 9661 |
| '6xxx' | 0.824 | 0.996 | 0.614 | 0.992 | 0.815 | 0.981 | 1311 |
| '7xxx' | 0.263 | 0.991 | 0.248 | 0.983 | 0.263 | 0.818 | 1844 |

As we can see, the results are quite promising and further implementations are made for the rest of the events. What follows the results for the subfamilies of the common procedures, which are the most sensitive legal ads to be decoded:

| Cutoff | 0.9 | 0.99 | 0.999 |
|--------|-----|------|-------|
| event_code | Precision | Recall | Precision | Recall | Precision | Recall | Support |
| '51xx' | 1 | 0.039 | 1 | 0.012 | 1 | 0.87 | 50000 |
| '52xx' | 1 | 0.084 | 1 | 0.072 | 1 | 0.774 | 50000 |
| '53xx' | 1 | 0.091 | 1 | 0.097 | 1 | 0.913 | 50000 |
| '54xx' | 1 | 0.09 | 1 | 0.097 | 1 | 0.841 | 50000 |
| '55xx' | 1 | 0.082 | 1 | 0.097 | 1 | 0.825 | 50000 |
| '56xx' | 1 | 0.097 | 1 | 0.092 | 1 | 0.979 | 50000 |

The results of the model are detailed in what follows. We have randomly selected 60 of the events in question due to data protection legislatures.

8) Discussion

We propose a multi-label Residual Convolutional Neural Network text classifier using Content-Based Routing process which train the model using content based orienting method described above. The benefits of our model is mainly in the optimization of the number of models needed to be trained in a later moment. With this method, we have succeeded in developing a highly accurate multi-classification model which will classify all of the 161 different events based on the text. One of the greatest difficulties is how to handle the variety and the veracity of the data which was available with different sizes. In order to overcome this problem, we have developed a bigger dataset containing as much types of events possible.

CONCLUSION

During this module, we developed a multi-label Residual Convolutional Neural Network text classifier using Content-Based Routing process to predict the final event code for each ad. We developed a decision tree to predict the code with the least possible false positives and the most accurate result possible. The tree was implemented using a content-based routing process, which allowed us to validate the model.
REFERENCES

[1] BODÉN, Mikael. A guide to recurrent neural networks and backpropagation. School of Information Science, Computer and Electrical Engineering, Halmstad University. November 13th, 2001.

[2] GRAVES, Alex; MOHAMED, Abdel-rahman; HINTON, Geoffrey. Speech Recognition with Deep Recurrent Neural Networks, 2013.

[3] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep Residual Learning for Image Recognition, December 10th, 2015.

[4] KHAN, Salman; RAHMANI, Hossein; ALI SHAH, Syed Afqaq; BENNAMOUN, Mohammed. A Guide to Convolutional Neural Networks for Computer Vision. Morgan and Claypool publishers, 2018.

[5] LAFFERTY, John; MCCALLUM, Andrew; PEREIRA, Fernando. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In International Conference on Machine Learning, 2001.

[6] NIELSEN, Michael, Using neural nets to recognize handwritten digits, Neural Networks and Deep Learning, June 2019.

[7] HOHPE, Gregor; WOOLF, Bobby. Enterprise Integration Patterns, October 10th, 2003, Princeton University.

[8] SERRA, Joan; KARATZOGLOU, Alexandros. Getting Deep Recommends Fit: Bloom Embeddings for Sparse Binary Input/Output Networks, June 13th, 2017.

[9] SHERSTINSKY, Alex. From RNN to Vanilla LSTM Network. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network, chapter 5, page 17, Directly Software Incorporated, August 9th, 2018.