Soccer news data collection, labelling and classification using machine learning

1st Aanund Jupskås Nordskog
Simula
University of Oslo
Oslo, Norway
aanundjn@ifi.uio.no

2nd Pål Halvorsen
SimulaMet
Oslo Metropolitan University
Oslo, Norway
paalh@simula.no

2nd Steven Hicks
SimulaMet
Oslo Metropolitan University
Oslo, Norway
steven@simula.no

3rd Håkon K. Stensland
Simula
Oslo, Norway
haakons@simula.no

3rd Hugo L. Hammer
Oslo Metropolitan University
Oslo, Norway
hugoh@oslomet.no

3rd Michael A. Riegler
SimulaMet
Oslo, Norway
michael@simula.no

***Abstract*—In the today's Internet we are overwhelmed with information. One important part of these information are news. Within news sport news are very popular in within sport news soccer takes a large amount of the coverage. For users it can be hard to find the really interesting news and it can even end up as a tedious and time consuming task. In this paper we present different machine learning approaches that are applied to soccer news from Norwegian news papers. We present a system to collect, index, label, analyse and present the collected news articles based on the content. In addition we compare different algorithms ranging from more traditional machine learning approaches to deep learning. In addition, the collected dataset is also made public available for other researchers.**

***Index Terms*—News, Soccer, Sport, Machine Learning.**

I. Introduction

The world wide web is almost an endless source of information, and a lot of this information comes in the form of text. From 2015 to 2018 the total number of websites have doubled, reaching 1.63 billion sites in 2018 [1]. With this amount of data available the need to categorize, index and label information is more important than ever.

Along with the increase in data the popularity of machine learning has also increased. The search term from google trends for machine learning and deep learning over the past five years shows that the popularity has grown significantly [2].

The popularity of deep learning is not without reason, for it has shown extraordinary results in many different application. In image classification CNN have shown high performance [3], Google translate is using recurrent neural networks (RNN) to translate from one language to another [4], and in natural language processing both RNN and convolutional neural networks (CNN) have seen a rise in popularity and state-of-the-art performance [5].

The main goal of this paper is to evaluate how deep learning and traditional machine learning methods perform on text classification, more specifically how the methods perform when classifying paragraphs in football articles. The goal is to train a wide range of deep learning and traditional machine learning models and see how they perform compared to each other. The models will be compared on classification performance, training time and setup complexity. Two methods will be used to determine the classification performance of the different models. The first method will compare the performance during training, i.e., metrics will be calculated during training and analysed. The second method will use the models created in method one and test how they perform in an application on new data. The application will present a visual result of how the models perform.

The data used for all experiments was extracted from VG.no and TV2.no (two large Norwegian newspapers), and paragraphs from these articles are labelled and stored in a database. An application is presented that was used to make the labelling of the paragraphs easier. In addition, an application is presented that displays a a visual representation of the paragraphs that have been classified by the different algorithms.

The research question of this work was: How does deep learning compare to traditional machine learning on soccer news classification when it comes to classification performance, training time and setup complexity? Thus, main contributions of this paper to provide an answer to the research question are:

- A thorough comparison between deep learning and traditional machine learning algorithms on text classification.
- A public dataset for football news collected from two newspapers that can be used to train and test machine learning algorithms.
- An application that can help to extend the dataset and provides a visual representation of how different models perform on the data.

II. Related Work

Natural language processing has seen state-of-the-art performance in many application over the last few years [5].

Dataset	c	l	N	$\lvert V\rvert$	Test
SST-1	5	18	11855	17836	2210
TREC	6	10	5952	9592	500

TABLE I: Summary of the dataset SST-1 and TREC. c: number of different classes. l: average number of word per sentence. N: Dataset size. $|V|$: Vocabulary size. Test: Test set size [8]

Much research has been tested on different text-based datasets with different deep learning and traditional machine learning models. We will see how the models mentioned above have performed on the Stanford Sentiment Treebank dataset (SST-1) and the TREC question dataset (TREC). There is a summary of the two datasets in table I.

The SST-1[1] dataset is a collection of movie reviews, and the objective of the classifier is to detect sentiment. There are five labels, very positive, positive, neutral, negative and very negative. One thing to note with this dataset is that the training set is provided at the parse-level, meaning that the training set consists of phrases and sentences. Therefore, the training set is an order of magnitude larger than what is shown in table I. The test set, on the other hand, consists only of sentences [6].

The TREC[2] dataset is a collection of questions, and the objective of the classifier is to classify the questions to different question types. For example, questions about location will classify to the location class. There are six labels, Abbrev., Entity, Description, Human, Location and Numeric. [7]

Table II shows the accuracy the different models have on the SST-1 and TREC datasets. The results are collected from different research papers that have used Naive Bayes, SVM, CNN and RNN on the two datasets.

The CNN-rand model is from Yoon Kim paper [8] where he tests different CNN models on different text datasets among them SST-1 and TREC. The CNN-rand model is the only model that does not use pre-trained vectors with word2vec. The vectors are instead randomly initialised and modified during training. This model is chosen because word2vec is beyond the scope of this thesis.

The Bi-LSTM model is used as a comparison to C-LSTM neural networks presented in Chunting Zhou paper [9], and it is tested on the SST-1 and TREC dataset. The Bi-LSTM is a one layered bidirectional recurrent network with the LSTM architecture.

Socher presents the SVM and Naive Bayes model used on the SST-1 in the paper [6] where he presents the SST-1 dataset. They are both implemented with bag-of-words features. Zhang presents the SVM and Naive Bayes model used on the TREC dataset in the paper [10] where he tests SVM on question classification. Both models are implemented with bag-of-words features and default values for the parameters (e.g. C value in SVM). The SVM model that is shown in the table use a linear kernel, but there were the same results for the polynomial, RBF and sigmoid kernel. Another thing to mention is that Silva [11] managed to get 95% accuracy on the TREC dataset with SVM. However, that implementation of the SVM model had highly engineered features.

[1]http://nlp.stanford.edu/sentiment

[2]http://cogcomp.cs.illinois.edu/Data/QA/QC/

Model	SST-1	TREC
Naive Bayes	41.0	77.4
SVM	40.7	85.8
BiLSTM	47.8	93.0
CNN-rand	45.0	91.2

TABLE II: Accuracy score for the different models on the SST-1 and TREC dataset

Article_id	content_order	content	html_type	class
yvdVOJ	0	To ml av Espen Ruud - og Sarpsborg gikk p sitt fjerde strake tap	h1	
yvdVOJ	1	(Sarpsborg-Odd 12) Odd-backen Espen Ruud (34) har laget fem ml p de	p	
yvdVOJ	2	Mens stfoldingene presterer med bravur i Europa, gr det alt annet	p	
yvdVOJ	3	Espen Ruud headet Odd i ledelsen, en annen back, Joachim Thomassen Ronaldo denne hsten).	p	

TABLE III: View of ow the data is stored in the database

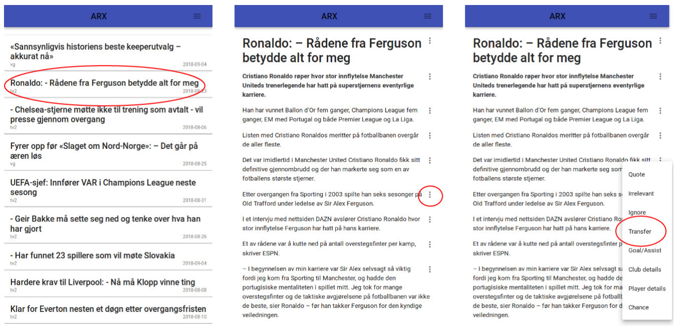
The results show that the deep learning models outperform the traditional machine learning models by a good margin. The bidirectional LSTM has the highest performance closely followed by the CNN model; there is a small jump down to SVM and Naive Bayes. They perform about the same on the SST-1 dataset, but the SVM model has higher performance on the TREC dataset.

III. Data

A. Retrieving Data

To collect the data we created a python program to fetch and process the HTML code from the web pages of VG and TV2. First, the program uses Request to fetch the HTML code of all the football articles on 'www.vg.no/fotball' and 'www.tv2.no/fotball'. Next, the relevant information from the HTML code is extracted using beautiful soup and the result is stored in the database. There are four columns in the database table containing the extracted information and one containing the class label (see table III):

- **article_id** - This is the id of the article given by TV2 and VG.
- **content_order** - This is the order of the paragraphs in the article. Zero is the headline, and the highest number is the last paragraph.
- **Content** - The content in the paragraph.
- **html_type** - The type of the HTML tag used by VG and TV2. For example p for paragraph and h1 for the headline.
- **class** - the label of the paragraph.



«Sannsynligvis historiens beste keeperutvalg – akkurat nå»
vg 2018-09-04

Ronaldo: - Rådene fra Ferguson betydde alt for meg
tv2 2018-08-23

- Chelsea-stjerne møtte ikke til trening som avtalt - vil presse gjennom overgang
tv2 2018-08-06

Fyrer opp før «Slaget om Nord-Norge»: – Det går på æren løs
vg 2018-08-25

UEFA-sjef: Innfører VAR i Champions League neste sesong
tv2 2018-08-31

- Geir Bakke må sette seg ned og tenke over hva han har gjort
tv2 2018-08-26

- Har funnet 23 spillere som vil møte Slovakia
tv2 2018-09-04

Hardere krav til Liverpool: - Nå må Klopp vinne ting
tv2 2018-08-08

Klar for Everton nesten et døgn etter overgangsfristen
tv2 2018-08-10

Ronaldo: – Rådene fra Ferguson betydde alt for meg

Cristiano Ronaldo røper hvor stor innflytelse Manchester Uniteds trenerlegende har hatt på superstjernens eventyrlige karriere.

Han har vunnet Ballon d'Or fem ganger, Champions League fem ganger, EM med Portugal og både Premier League og La Liga.

Listen med Cristiano Ronaldos meritter på fotballbanen overgår de aller fleste.

Det var imidlertid i Manchester United Cristiano Ronaldo fikk sitt definitive gjennombrudd og der han markerte seg som en av fotballens største stjerner.

Etter overgangen fra Sporting i 2003 spilte han seks sesonger på Old Trafford under ledelse av Sir Alex Ferguson.

I et intervju med nettsiden DAZN avslører Cristiano Ronaldo hvor stor innflytelse Ferguson har hatt på hans karriere.

Et av rådene var å kutte ned på antall overstegsfinter per kamp, skriver ESPN.

– I begynnelsen av min karriere var Sir Alex selvsagt så viktig fordi jeg kom fra Sporting til Manchester, og hadde den portugisiske mentaliteten i spillet mitt. Jeg tok for mange overstegsfinter og de taktiske avgjørelsene på fotballbanen var ikke de beste, sier Ronaldo – før han takker Ferguson for den kyndige veiledningen.

Ronaldo: – Rådene fra Ferguson betydde alt for meg

Cristiano Ronaldo røper hvor stor innflytelse Manchester Uniteds trenerlegende har hatt på superstjernens eventyrlige karriere.

Han har vunnet Ballon d'Or fem ganger, Champions League fem ganger, EM med Portugal og både Premier League og La Liga.

Listen med Cristiano Ronaldos meritter på fotballbanen overgår de aller fleste.

Det var imidlertid i Manchester United Cristiano Ronaldo fikk sitt definitive gjennombrudd og der han markerte seg som fotballens største stjerner.

Etter overgangen fra Sporting i 2003 spilte han seks s Old Trafford under ledelse av Sir Alex Ferguson.

I et intervju med nettsiden DAZN avslører Cristiano Ro stor innflytelse Ferguson har hatt på hans karriere.

Et av rådene var å kutte ned på antall overstegsfinter p skriver ESPN.

– I begynnelsen av min karriere var Sir Alex selvsagt s fordi jeg kom fra Sporting til Manchester, og hadde de portugisiske mentaliteten i spillet mitt. Jeg tok for ma overstegsfinter og de taktiske avgjørelsene på fotballb de beste, sier Ronaldo – før han takker Ferguson for d veiledningen.

Fig. 1: Example of how the labelling is done with Arx. The first step is to click an article, the second step is to locate a paragraph, and the third step is to determine which label to give the chosen paragraph.

The database contains approximately 1000 articles and 20,000 paragraphs and 1,000 labelled paragraphs from each class.

B. Labelling Data

After the data was stored in the database, we created a web app called Arx to label the data. The purpose of Arx was to help with the labelling, making the process much quicker. Arx displays the content of each article in a structured way with a labelling option for each paragraph (see figure 1). It is also possible to fetch all paragraphs or articles containing keywords (see figure 2). For example, one can fetch all articles or paragraphs containing the word ”mlscorer”, ”overgang” or ”vinnermlet”. This option is beneficial since there are a lot of football articles that do not contain relevant paragraphs. Labelling is possible on both computers and mobile devices.

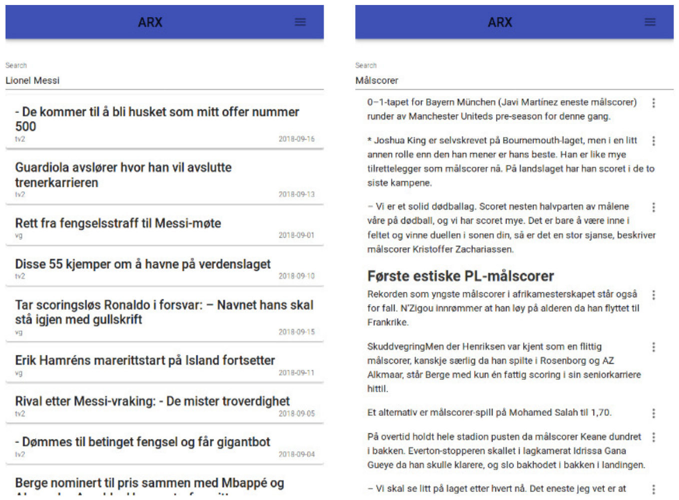
Arx was made with Django as the RESTful API server and React as the frontend web page. There are to possible HTTP commands to the Django API, GET and PUT. React calls GET to fetches articles and paragraphs from the database, and PUT to update a paragraph with the correct label in the database. There are in general only five labels, however, to make the labelling easier there are several more option to choose from than just five, for example, club detail, player detail and chances to name a few.

C. The Dataset

The dataset contains 5,526 labelled data samples. The dataset can be found online here [12]. In table IV there is an overview of how many data samples there are of each class. There are eight classes, but only five will be used because the last three classes have too few samples. The dataset is split with satisfied 10-fold cross-validation during training.

IV. Experiments

The football articles are collected from Norwegian newspapers, and there are no libraries for pre-processing Norwegian text. As a result, there will be no pre-processing of the text in the dataset, except for removing special characters.



Lionel Messi

- De kommer til å bli husket som mitt offer nummer 500
tv2 2018-09-16

Guardiola avslører hvor han vil avslutte trenerkarrieren
tv2 2018-09-13

Rett fra fengselsstraff til Messi-møte
vg 2018-09-01

Disse 55 kjemper om å havne på verdenslaget
tv2 2018-09-10

Tar scoringsløs Ronaldo i forsvar: – Navnet hans skal stå igjen med gullskrift
vg 2018-09-15

Erik Hamréns marerittstart på Island fortsetter
vg 2018-09-11

Rival etter Messi-vraking: - De mister troverdighet
tv2 2018-09-05

- Dømmes til betinget fengsel og får gigantbot
tv2 2018-09-04

Berge nominert til pris sammen med Mbappé og

Målscorer

0–1-tapet for Bayern München (Javi Martínez eneste målscorer) runder av Manchester Uniteds pre-season for denne gang.

* Joshua King er selvskrevet på Bournemouth-laget, men i en litt annen rolle enn den han mener er hans beste. Han er like mye tilrettelegger som målscorer nå. På landslaget har han scoret i de to siste kampene.

– Vi er et solid dødballag. Scoret nesten halvparten av målene våre på dødball, og vi har scoret mye. Det er bare å være inne i feltet og vinne duellen i sonen din, så er det en stor sjanse, beskriver målscorer Kristoffer Zachariassen.

Første estiske PL-målscorer

Rekorden som yngste målscorer i afrikamesterskapet står også for fall. N'Zigou innrømmer at han løy på alderen da han flyttet til Frankrike.

SkuddvegringMen der Henriksen var kjent som en flittig målscorer, kanskje særlig da han spilte i Rosenborg og AZ Alkmaar, står Berge med kun én fattig scoring i sin seniorkarriere hittil.

Et alternativ er målscorer-spill på Mohamed Salah til 1,70.

På overtid holdt hele stadion pusten da målscorer Keane dundret i bakken. Everton-stopperen skallet i lagkamerat Idrissa Gana Gueye da han skulle klarere, og slo bakhodet i bakken i landingen.

– Vi skal se litt på laget etter hvert nå. Det eneste jeg vet er at

Fig. 2: Example of how to fetch articles and paragraphs with a keyword in Arx. The left figure finds all articles that contain ”Lionel Messi”. The right figure finds all paragraphs that contain the word ”Mlscorer”

Class Label	Count
Goal/Assist	1117
Quotes	975
Transfer	887
Irrelevant	812
Ignore	663
Player Detail	340
Club Detail	315
Chances	300

TABLE IV: Number of data samples per class.

In table V there is an overview of the hardware used to train the different machine learning algorithms.

The traditional machine learning algorithms Navie Bayes, Linear SVC and SVC are all created with the Scikit-learn library. With Scikit-learn the algorithm is imported and initiated with the relevant hyperparameters. All the algorithms choose between two vectorizers to transform the paragraphs into vectors of numbers. The first method is term frequency-inverse document frequency (TFIDF), and the other method is a count vectorizer.

The deep learning models RNN and CNN are both created with the python library Keras. Keras [13] is a high-level neural network API written in python, that runs on top of Tensorflow

Component	Model	Description
CPU	Intel i7-8700K [3]	Cores: 6 Clock spees: 3.7-4.7GHz Catch: 12MB
GPU	Nvidia GTX1070 Ti [4]	Memory: 8GB DDR5 Cuda cores: 2432 Boost clock: 1683 MHz
RAM	na	16 GB DDR4
OS	Ubuntu 18.04	na

TABLE V: Hardware specifications

[14], Theano [15] or CNTK [16]. We use the tokeniser from Keras to convert words to numbers. The input to the models is fixed, and the paragraph with the most words determine the input length. All other paragraphs are padded to match the max size.

The RNN model is a composition of the following layers, embedding layer, dropout layer, bidirectional LSTM layer, and a dense layer. To find the optimal RNN model three experiments will be executed. The first experiment will find the optimal embedding dimension for four different LSTM layers, with the other parameters set to the default values. For the second experiment, the optimal embedding dimension will be used to find the optimal number of LSTM neurons in the LSTM layer, with the other parameters set to the default value. The third experiments will use the optimal embedding dimension and the optimal number of LSTM neurons, to find the optimal dropout rate when the dropout layer is before and after the LSTM layer.

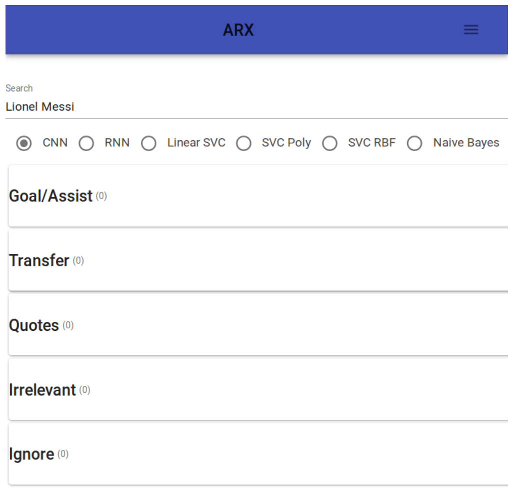
The CNN model is a composition of these layers, embedding layer, convolutional layer, dropout layer, pooling layer, flattening layer and two dense layers.

To find the optimal CNN model four experiments will be executed. The first experiment will find the optimal embedding dimension with different filter sizes, pooling sizes and kernel sizes. The second experiment will use the optimal embedding dimension, kernel size and pooling size to find the optimal filter size. The third experiment will use the optimal values found in experiment one and two to find the optimal number of neurons in the dense layer. The fourth experiment will find the optimal dropout rate when the optimal values are used for all parameters.

After the training experiments, each model will use the optimal parameters and train on the entire dataset. The models will then be used in the application to classify articles about football players. The result from the models will be compared and analysed. The articles used in the application is collected in the period between December 2018 and February 2019, three months after the training set was made.

Figure 3 shows the user interface of the classification program. The application takes a player name and a machine learning type as input. It then searches the database for all articles where that player is mentioned two times. The paragraphs in these articles are run through the machine learning algorithm, and the result is presented in five different tags shown in figure 3.

Each model will classify articles for five players. Two players will test the models on how well they classify ”Goal/Assist” paragraphs. The two players are Lucas Moura and Marcus Rashford, both of these players scored goals and performed well in the period from December 2018 to February 2019. Two players will test the models on how well they classify ”Transfer” paragraphs. The two players are Higuain and Morata, these players were sold in the 2019 January transfer window. The last player to test the models is Martin degaard, he performed well and was speculated in the transfer marked in the period December 2018 to February 2019.


ARX

Search
Lionel Messi

[✓ CNN] RNN Linear SVC SVC Poly SVC RBF Naive Bayes

Goal/Assist (0)

Transfer (0)

Quotes (0)

Irrelevant (0)

Ignore (0)

Fig. 3: An example of the classification of Lionel Messi with the CNN model. Each paragraph in articles about Lione Messi will be classified in one of the five tabs.

Metric values and confusion matrix are calculated for each player on all the different models. In addition, for Lucas Moura and Morata the paragraphs will be shown and the result will be analysed. Five articles will be classified for Lucas Moura.

The following metrics are used to determine the performance of a classifier: Accuracy, recall, precision, f-measure and MCC. To get the result of the classifiers as general as possible the metrics are calculated as an average over 10-fold cross-validation.

A. Results

The performance of Linear SVC, SVC RBF and SVC poly are almost identical during training, with an MCC of 82% and an accuracy of 85.5%. The Naive Bayes model performs 4% lower than the others. All the algorithms have the same pattern, where the f1 score of the ”Irrelevant” class has low performance, and the other classes have high performance. Especially for Naive Bayes the f1 score of the ”Irrelevant” class is 17% lower than the other algorithms, but for the other four classes, the performance is only 2% lower. The training time for SVC poly and SVC RBF is 2.7 seconds, that is 30 times higher than what it is for Naive Bayes and Linear SVC. And the prediction time is 0.4 ms per paragraph, and that is 25 times higher than what it is for Linear SVC and Naive Bayes.

When running the optimal model for all the algorithms in the application, the results show a 20% drop in MCC for all the models. The SVC poly and SVC RBF have the best performance, with an MCC of 62%, beating Linear SVC with 2% and Naive Bayes with 6%. Then Naive Bayes classifier has a 5-7% higher f1 score on the ”Transfer” class than the others. On the other hand, the f1 score of the ”Ignore” class is 13-15% lower than the other models.

The performance of the RNN and CNN classifier is very similar during training. The RNN model has an MCC of 85.3% and an accuracy of 88.3%, which is 1% better than CNN. The

Parameter	Metrics					f1 score for each class				
Model Name	Acc	MCC	Rec	F1	Prec	Go/As	Tr	Qu	Ir	Ig
RNN	0.887	0.859	0.887	0.885	0.892	0.907	0.884	0.982	0.74	0.912
CNN	0.879	0.849	0.879	0.876	0.882	0.9	0.879	0.979	0.732	0.888
SVM RBF	0.859	0.823	0.859	0.857	0.864	0.903	0.876	0.903	0.701	0.9
SVM Poly	0.859	0.822	0.859	0.855	0.861	0.904	0.882	0.903	0.694	0.891
Linear SVM	0.853	0.815	0.853	0.848	0.854	0.907	0.877	0.893	0.68	0.883
Naive Bayes	0.818	0.775	0.818	0.798	0.814	0.873	0.868	0.87	0.528	0.851

TABLE VI: Shows the result of the training experiments

Parameter	Metrics					f1 score for each class				
Model Name	Acc	MCC	Rec	F1	Prec	Go/As	Tr	Qu	Ir	Ig
RNN	0.733	0.647	0.733	0.717	0.778	0.611	0.672	0.936	0.718	0.65
CNN	0.703	0.625	0.703	0.696	0.776	0.558	0.684	0.881	0.699	0.659
SVM Poly	0.703	0.624	0.703	0.72	0.768	0.567	0.661	0.868	0.667	0.835
SVM RBF	0.697	0.618	0.697	0.711	0.767	0.563	0.65	0.872	0.663	0.809
Linear SVM	0.69	0.609	0.69	0.704	0.758	0.566	0.633	0.85	0.657	0.813
Naive Bayes	0.613	0.564	0.613	0.645	0.738	0.515	0.711	0.855	0.467	0.677

TABLE VII: Shows the result of the application experiments

Model Name	Expression	ttt
Naive Bayes	$4 * 80 * 0.063 * 10$	3 min
Linear SVM	$3 * 80 * 0.084 * 10$	3.36 min
SVM RBF	$4 * 80 * 2.7 * 10$	2.4 h
SVM Poly	$6 * 80 * 2.6 * 10$	3.46 h
RNN	$7 * 80 * 11 * 10$	17 h
CNN	$27 * 80 * 8 * 10$	48 h

TABLE VIII: Show the total training time for each algorithm

two models have the same pattern, where the f1 score of the ”Irrelevant” class is low, and the other classes f1 score is high. The training time for CNN is between 4 and 11 seconds, and for RNN it is between 7 and 15 seconds. The predictions time for CNN is 0.22 ms per paragraphs, which is about half of what it is for RNN.

When running the optimal model for RNN and CNN in the application, the results show a 20% drop in MCC score. The RNN classifier has the best performance, with an MCC of 64.7%, which is 2% higher than what it is for CNN. The f1 score of the different classes is very similar between the two models.

V. Discussion

The performance of the algorithms during training was overall very high. There is an overview of the top results in table VI. Naive Bayes is at the low end with an MCC of 75% and RNN at the high end with an MCC of 85.3%. The MCC of the deep learning models was around 85%, while for the SVM models the MCC was around 82%. Thus, the performance is overall better for the deep learning algorithms, but not with a large margin. After the training, each model was used in the application with the optimal parameters. The first thing to note from the results in the application is that all the algorithms suffered a $\sim 20\%$ drop in MCC. This drop might indicate that the models are overfitting, or that the training set does not represent the data good enough. Since all the algorithms are affected the same, the problem most likely lies with the training set. The results from the application show that the deep learning models have a better performance than the traditional machine learning models; there is an overview of the results in table VII. Same as during training, RNN has the highest MCC with 64.7%, and Naive Bayes has the lowest MCC with 56.4%. The SVM model with polynomial kernel was closest with an MCC of 62.4%. The difference in performance is marginal, with RNN performing slightly better than SVM models.

The training time can be divided into three brackets, where Naive Bayes and linear SVM have a very low training time, SVM with polynomial and RBF kernel have a high training time, and RNN and CNN have a very high training time. In table VIII there is an overview of the approximate time for the different models. Where the time is calculated with the function 1 and where the different parameters are as follow:

- Experiments (exp) - Number of experiments that were executed.
- Parameters per experiment (ppe) - Number of values that were tested for each experiment.
- Average training time (att) - The average training time for each model.
- Number of folds in cross-validation (fcv) - Number of folds that were used during training.

$$Total\ training\ time\ (ttt) = exp * ppe * att * fcv \qquad (1)$$

SVM with RBF and polynomial kernel have a lower training time then what is shown in table VIII. The implementation from Sklearn does not support multi-threading, so each of them only ran on one processor core. However, during training, the task was split up to use the six cores available. Therefore, the time shown in the table can roughly be divided by six. The linear SVM model has a lower training time than the other SVM models because Sklearn uses a different multi-class implementation. For linear SVM Sklearn uses one-vs-rest, and for the other two, it uses one-vs-one, which is more computationally expensive. The total training time is much higher for the deep learning algorithms than it is for the traditional machine learning models. One reason for the high training time is that there are more parameters to optimise for the deep learning models, combined with higher training time in general. Despite the high training time compared to the other models, it is not unreasonable high and still very manageable.

A. *Failure Analysis*

The algorithms have some common performance problems. The ”Irrelevant” class have a low f1 score compared to the other classes. Moreover, paragraphs classify to the ”Goal/Assist” class instead of the ”Irrelevant” class. There is also a significant drop in performance from the training results to the application results.

The ”Irrelevant” class contains many different types of paragraphs, for example, goal chances, team lineups and general trivial information. In other words, the ”Irrelevant” class is many classes put into one class, and this makes it difficult for the models to generalise what belongs to this class. One solution to increase the performance of the different models can be to change the ”Irrelevant” class. For example, one of the main problems in the application is that paragraphs about goal chances get classified to the ”Goal/Assist” class instead of to the ”Irrelevant” class. Renaming the ”Goal/Assist” class to ”Situation” class, and changing the content from being about goals and assists to goals, assists and chances would simplify the ”Irrelevant” class. Another problem in the application is that paragraphs about team line-ups get classified to the ”Ignore” class instead of to the ”Irrelevant” class. Moving these paragraphs to the ”Ignore” class makes sense, maybe that is where they should have been from the start. These two measures would reduce the size of the ”Irrelevant” class without complicating the ”Goal/Assist” and ”Ignore” class.

One possible reason for the big difference between the training experiment result and the application result can be the labelling method used on the training set and the test set. There are three ways to label data samples (1) label paragraphs in random articles, (2) label paragraphs in articles that contain a keyword, for example, articles that contain a certain player name, or (3) label paragraphs from a list of paragraphs containing a keyword, for example, a list of all paragraphs that contain the word ”Mlscorer”. When making the training set, all three methods were used, but the third method was used the most. On the other hand, when making the test set, only the second method was used, which means that the training set has a less diverse set of paragraphs compared to the test set.

RNN and CNN are data hungry algorithms compared to Naive Bayes and SVM, and the training set used in this research contains approximately 5,500 data samples, which is considered to be few when it comes to deep learning. However, even with a limited training set, the deep learning algorithms outperformed the traditional machine learning algorithms, and if this dataset is a working progress, meaning that it will grow over time, it is reasonable to think that CNN and RNN will benefit more from this than SVM and Naive Bayes.

VI. Conclusion and Future Work

The research question raised was: ”**How does deep learning compare to traditional machine learning on text classification when it comes to classification performance, training time and setup complexity?**”. To answer this question different machine learning and deep learning models were created, and then compared against each other.

There is no clear answer to whether deep learning or traditional machine learning algorithms are preferred on text classification, based on the research done in this paper. On the one hand, the deep learning models did perform slightly better overall compared to the traditional machine learning algorithms. However, on the other hand, for the deep learning models, the training time is much higher and the setup is more challenging.

To conclude, if the dataset is limited as it is in this work the traditional machine learning models are the better choice. The easy setup, low training time and limited maintenance outweigh the small performance gain given by deep learning. However, if the dataset is growing over time and have the potential to become much bigger than it is now, the deep learning models are the better choice. The overhead would then be worth the potential performance gain that is given by deep learning.

For future work we are planing the following improvements. First, to get a more thorough comparison between deep learning and traditional machine learning, more algorithms can be tested. For example, add decision trees, k-nearest neighbours and other machine learning algorithms. Also, test different architectures for RNN and CNN. Furthermore, pre-processing of the dataset was almost non-existent in this research; only special characters were removed. Therefore, more advanced pre-processing could be tested on the dataset. One way to do this is to use the Google Translate API, and translate the dataset into English for so to apply pre-processing tools available in English. Second, improve the dataset, by adding more data samples. See how the different models perform when given fewer data samples compared to more. Third, We plan to make the application so general that the only thing it needs is an unlabeled dataset, and eventually machine learning models to classify data samples.

References

[1] Total number of websites - internet live stats. [Online]. Available: http://www.internetlivestats.com/total-number-of-websites/

[2] Google trends. [Online]. Available: https://trends.google.com/trends/explore?date=today

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] G. Lewis-Kraus, “The great ai awakening,” *The New York Times Magazine*, vol. 14, 2016.

[5] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[6] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[7] X. Li and D. Roth, “Learning question classifiers,” in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.

[8] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[9] C. Zhou, C. Sun, Z. Liu, and F. Lau, “A c-lstm neural network for text classification,” *arXiv preprint arXiv:1511.08630*, 2015.

[10] D. Zhang and W. S. Lee, “Question classification using support vector machines,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003, pp. 26–32.

[11] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, “From symbolic to sub-symbolic information in question classification,” *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137–154, 2011.

[12] A. Nordskog, “Text classification project,” original-date: 2019-05-16T06:12:33Z. [Online]. Available: https://github.com/Halflingen/Text-Classification-Project

[13] F. Chollet *et al.*, “Keras,” 2015. [Online]. Available: https://keras.io/

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: a system for large-scale machine learning.” in *OSDI*, vol. 16, 2016, pp. 265–283.

[15] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: A cpu and gpu math compiler in python,” in *Proc. 9th Python in Science Conf*, vol. 1, 2010.

[16] F. Seide and A. Agarwal, “Cntk: Microsoft’s open-source deep-learning toolkit,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2135–2135.