Beyond Transformers: fault type detection in maintenance tickets with Kernel Methods, Boost Decision Trees and Neural Networks

Beyond Transformers: fault type detection in maintenance tickets with Kernel Methods, Boost Decision Trees and Neural Network

S. Campese[1,2,3] **F. Agostini[1,2,4]** J. Pazzini[1,2] D. Pozza[3]

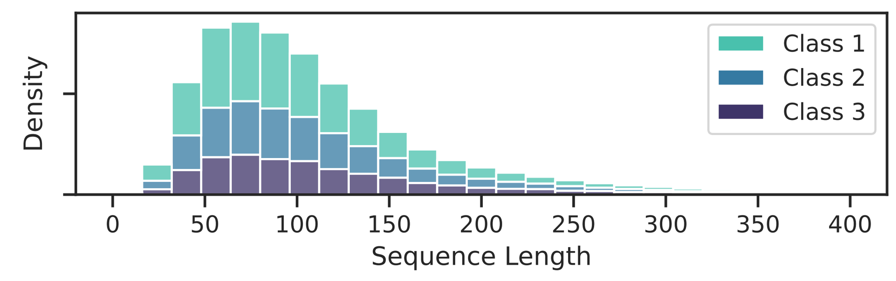
[1]Department of Physics and Astronomy, University of Padua
[2]National Institute for Nuclear Physics, Padua Division
[3]R&D Department, Omnys SRL
[4]Department of Control and Computer Engineering, Polytechnic University of Turin

Introduction

The proper handling of **customer tickets and maintenance requests** is pivotal for enterprises. It directly impacts customer satisfaction and consequentially it leads to **higher economic and brand-image revenues**. Several methods based on **Natural Language Processing** (NLP) have been developed to classify, tag, and prioritize customer support requests and maintenance tickets. However, the **specific domain** of each company, in conjunction with the different products and services offered, make it **difficult to develop generalized solutions**.

Purpose

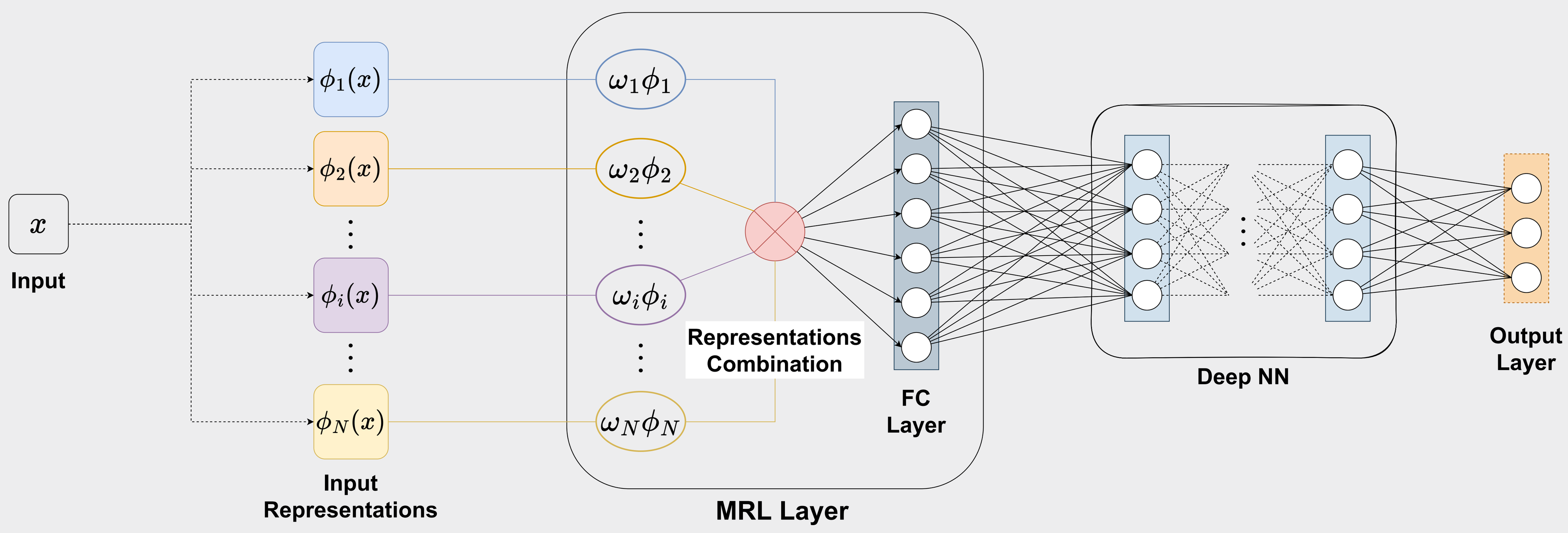
In this work, we propose two approaches to predict the type of fault from the text of maintenance support tickets: (i) Kernel Methods in conjunction with Boost Decision Trees (***Spectrumboost***), and (ii) Neural Network for Multiple Representation Learning (***DeepMRL***). Those models are tested and compared against state-of-the-art solutions based on Transformers architectures on a **real-world set of 131305 tickets in the Italian language**. Results suggest that the **proposed models outperform Transformers** both in the prediction **accuracy** and in the **time and computational resources** required for their training.



Dataset	Class 1	Class 2	Class 3	Avg. Seq. length
Training	42885	29246	22378	$59.87_{\pm 39.47}$
Validation	4659	3378	2465	$59.15_{\pm 39.40}$
Test	11888	8113	6293	$59.92_{\pm 38.88}$
Total	59432 (45.3%)	40737 (31.0%)	31136 (23.7%)	$59.82_{\pm 39.34}$

Maintenance tickets sequence length and division in training, validation and test set.

Deep Multiple Representation Learning

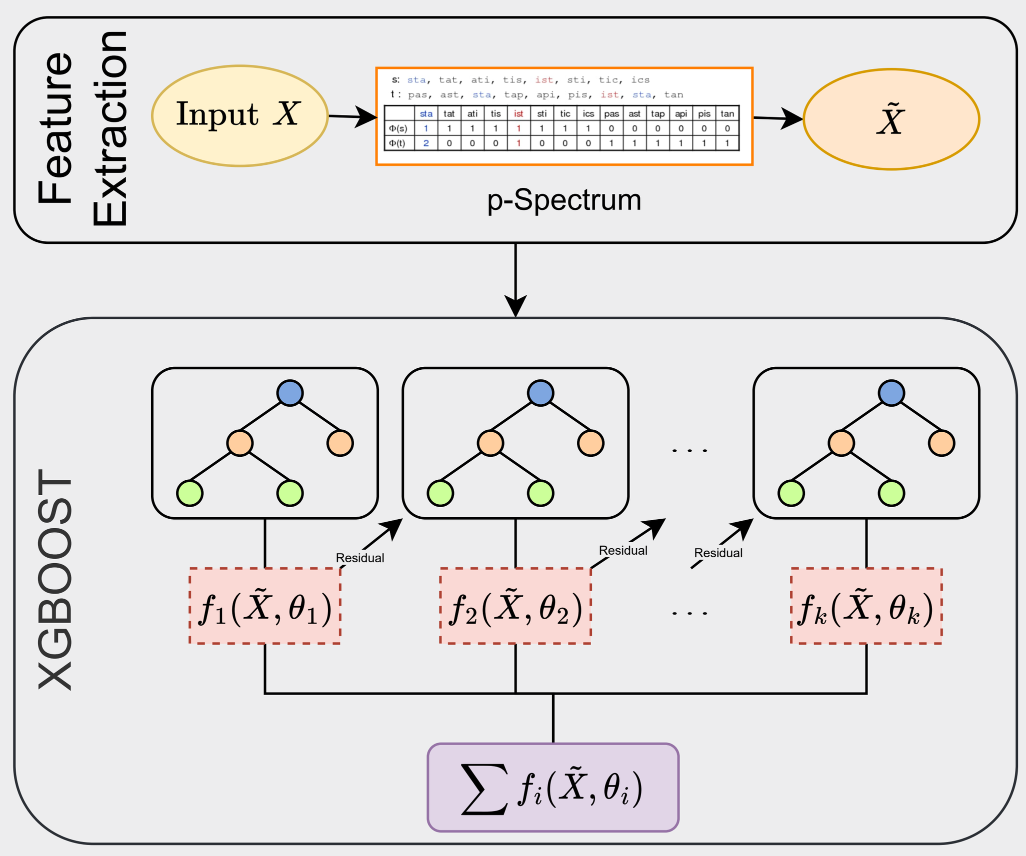


The novel **Multiple Representation Learning** (MRL) layer mimics the logic behind Multiple Kernel Learning by learning a **new data representation** X_{comb} as a **linear or non-linear combination of base reprentations** $\phi_i(x)$, where ϕ_i can be an arbitrary function (also a kernel one), a BERT encoding, a NN embedding, or other.

By applying **constraints to the learned weights** ω_i, it is possible to compute different type of combinations, such as Convex and Affine approaches.
The new representation is then passed through a **fully-connected layer** with L1 and L2 regularization to **learn non-linear dependencies**.

SpectrumBoost

SpectrumBoost extracts features from text using the **p-Spectrum kernel** with **Nyström Approximation**. This kernel counts any possible contiguous sub-sequence of length p and it focuses on local information. These are fed into an **XGBoost classifier**, that provides the label.



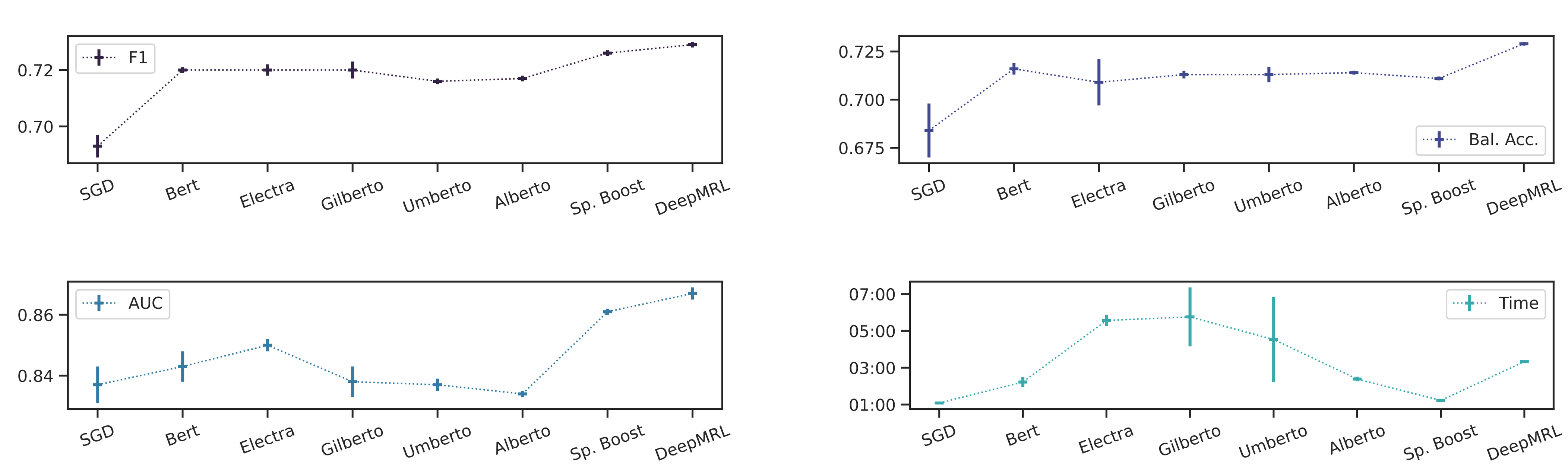
Results

Experiments have been conducted using 2x Nvidia GTX 1070, with the exception of 2x Nvidia V100 for Transformer-based models.

Transformers outperform the SGD baseline, but they require a large training time even employing high-performance Nvidia V100 GPUs.

Our **SpectrumBoost** is able to **surpass in performance all the models based on Transformers**, achieving a gain of 2.8% with respect to SGD and 1.1% with respect to the Electra architecture in terms of AUC. It also **outperforms the other models in terms of F1**, with a time comparable with SGD.

The newly proposed **DeepMRL outperforms all other models in all the considered metrics**. In terms of AUC, DeepMRL shows a 3% improvement compared to our baseline and 1.7% with respect to Electra. Considering the other metrics, DeepMRL outperforms the baseline and Transformer models, leading to an overall **gain of 0.9% for F1 score**, and **1.3% for Balanced Accuracy**.



Algorithm	F1 score	Balanced Accuracy	AUC score	Time
SGD ($p=5$)	$0.693_{\pm 0.004}$	$0.684_{\pm 0.014}$	$0.837_{\pm 0.006}$	1:05:03 ± 0:00:04
BERT	$0.720_{\pm 0.001}$	$0.716_{\pm 0.003}$	$0.843_{\pm 0.005}$	2:13:25 ± 0:16:07
Electra	$0.720_{\pm 0.002}$	$0.709_{\pm 0.012}$	$0.850_{\pm 0.002}$	5:33:56 ± 0:18:39
Gilberto	$0.720_{\pm 0.003}$	$0.713_{\pm 0.002}$	$0.838_{\pm 0.005}$	5:45:38 ± 1:36:04
Umberto	$0.716_{\pm 0.001}$	$0.713_{\pm 0.004}$	$0.837_{\pm 0.002}$	4:31:53 ± 2:18:46
Alberto	$0.717_{\pm 0.001}$	$0.714_{\pm 0.001}$	$0.834_{\pm 0.001}$	2:23:08 ± 0:07:26
SpectrumBoost ($p=4$)	$0.726_{\pm 0.001}$	$0.711_{\pm 0.001}$	$0.861_{\pm 0.001}$	**1:13:23** ± 0:00:08
DeepMRL ($p=[4,5,6]$)	$\mathbf{0.729}_{\pm 0.001}$	$\mathbf{0.729}_{\pm 0.001}$	$\mathbf{0.867}_{\pm 0.002}$	3:19:48 ± 0:00:27