Inferring the source of official texts: can SVM beat ULMFiT?

Pedro Henrique Luz de Araujo[1] Teófilo Emidio de Campos[1]
Marcelo Magalhães Silva de Sousa[2]

[1]Department of Computer Science, University of Brasília, Brasília – DF, Brazil

[2]Tribunal de Contas do Distrito Federal, Zona Cívico-Administrativa, Brasília – DF, Brazil

teodecampos@unb.br

Sponsored by FAPDF and CNPq

PROPOR 2020

Paper and code available at
`https://cic.unb.br/~teodecampos/KnEDLe/propor2020/`

Overview

Introduction

Motivation

- Government Gazettes are a great source of information of public interest:
  - Nominations, contracts, public notices
  - Public expenditures may be subject to frauds and irregularities
- Difficulties:
  - Unstructured data
  - Domain-specific language (official texts)

Examples

Excerpt 1

O GOVERNADOR DO DISTRITO FEDERAL, no uso das atribuições que lhe confere o artigo 100, incisos XXVI e XXVII, da Lei Orgânica do Distrito Federal, resolve [...]

Excerpt 2

Presidente da COVED, acolhendo os pareceres inseridos nos processos abaixo, declara habilitados para a venda à PRAZO os itens a seguir: [...]

Excerpt 2

[...] TORNAR PÚBLICO o resultado das investigações constantes nos processos dos servidores listados abaixo e que se configuraram em acidente de serviço, sem dano, nos termos do artigo 23, § 1º, inciso IV, do Decreto nº 34.023, de 10 de dezembro de 2012, observando-se a seguinte ordem: número do processo, nome e matrícula. [...]

Objectives

- Identify the institution of origin of documents fom the Official Gazette of the Federal District (DO-DF)
  - Information indexing
  - Public auditing
  - The use of rules and regular expressions is not robust
- Deal with limited labelled training set using transfer learning

Contributions

- A corpus of labelled and unlabelled Official Gazette documents
- Baseline evaluations with
  - Deep learning (ULMFiT) and
  - Shallow learning (BoW with Naïve Bayes and SVM)
- Ablation analysis of ULMFiT steps

The dataset

The dataset

- 2,652 texts extracted from the Official Gazette of the Federal District[1]
- Handcrafted regex rules to extract
  - publication date
  - section number and title
  - public body that issued the document
  - etc.
- 797 texts manually examined: 724 free of labelling mistakes.

[1] Available at https://www.dodf.df.gov.br/.

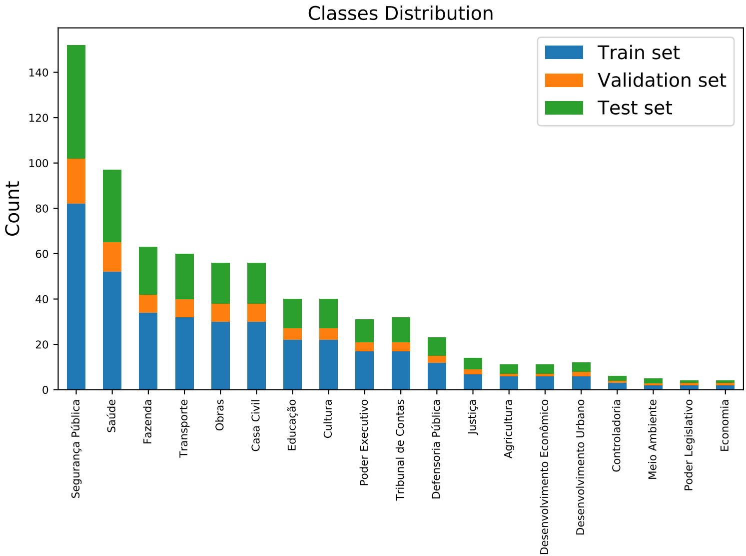
The dataset

- Documents originated from 25 different government entities
- Filter out entities with less than 3 samples, result:
  - 717 labelled examples (texts)
  - 19 classes (government entities that author the texts)
- Divide the data into two separate parts:
  - 717 labelled examples classification
  - 1,928 unverified or incorrectly labelled samples for unsupervised training of a language model

Classification data

- Hold-out method with
  - 384 (8/15) of the texts for the training set,
  - 96 (2/15) for the validation set and
  - 237 (5/15) for the test set
- Imbalanced data:
  - most frequent class (*Segurança Pública*) with 140 samples
  - least frequent classes with less than 5 documents

Data Distribution



Language model data

Standard language modelling task:

Label of each token is the following token in the sentence

LM dataset:

- Drop 2 empty texts, totalising 1,926 documents with 984,580 tokens
- Splits:
  - 784,260 tokens for training (80%)
  - 200,320 for validation (20%)
  - No test set: no need for unbiased evaluation of the language model

The methods

Preprocessing

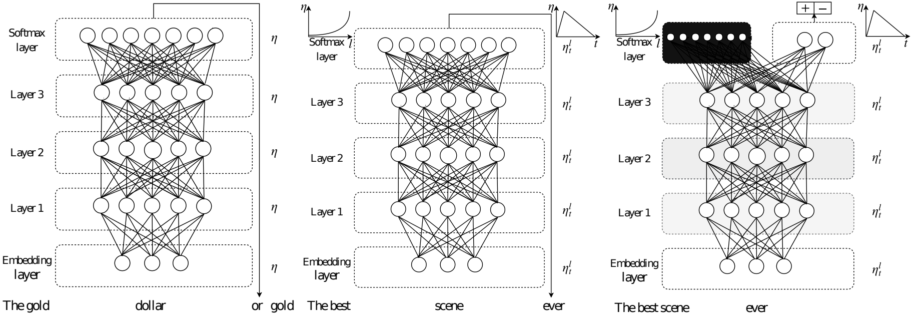
- Lowercase text and use SentencePiece [Kudo and Richardson, 2018] to tokenize.
  - The same used for the pretrained language model
- Add special tokens for padding, first letter capitalization, all letters capitalization, character and word repetition etc.[2]
- Final vocabulary of 8,552 tokens, including words, subwords, special tokens and punctuation.

[2] List of special tokens used available at `https://docs.fast.ai/text.transform.html`

Baseline

- Two kinds of BOW:
  - tf-idf values;
  - token counts.
- Two classifiers:
  - Naïve Bayes (NB);
  - Support Vector Machines (SVM) with linear kernel (a.k.a. w/o kernels).

Universal Language Model Fine-Tuning (ULMFiT) [Howard and Ruder, 2018]



(a) LM pre-training (b) LM f ne-tuning (c) Classif er f ne-tuning

Transfer learning

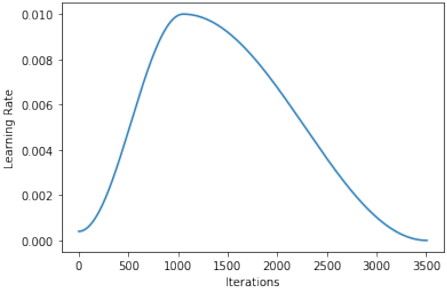
Stages:

- **Language model pre-training**: we use a bidirectional Portuguese language model[3] trained on 166,580 Wikipedia articles (100M tokens).
  - Architecture:
    - ★ 400-dimensional embedding layer
    - ★ 4 Quasi-Recurrent Neural Network layers (QRNN [Bradbury et al., 2016])
    - ★ linear classifier.

[3] Available at https://github.com/piegu/language-models/tree/master/models.

Transfer learning

- **Language model fine-tuning**: we fine-tune the forward and backward pre-trained language models on our unlabelled dataset.
  - We use discriminative fine-tuning (different learning rates for different layers) and cyclical learning rates [Smith and Topin, 2017] with cosine annealing to speed up training.



Transfer learning

- **Classifier fine-tuning**: we add two linear blocks to the language models (batch normalization [Ioffe and Szegedy, 2015] + dropout [Srivastava et al., 2014] + FC layer).
  - Final prediction is the average of the forward and backward models.
  - Let $\mathbf{h}_t$ be the hidden state of the last time step, and $\mathbf{H} = \{\mathbf{h}_{t-T}, \cdots, \mathbf{h}_t\}$, be the hidden states of as many time steps as can be fit in GPU memory. Then, the input to the linear blocks $\mathbf{h}_c$ is:

$$\mathbf{h}_c = \text{concat}(\mathbf{h}_t, \text{maxpool}(\mathbf{H}), \text{averagepool}(\mathbf{H})) \,. \tag{1}$$

Experiments

Baseline

- Random search + evaluation on validation set to find best hyperparameter values.
- Four scenarios:
  - tf-idf + NB;
  - tf-idf + SVM;
  - counts + NB;
  - counts + SVM.
- For each scenario we train 100 models with different randomly assigned hyperparameter values.
- tf-idf gave better results.

Transfer learning

- We use the learning rate range test [Smith, 2015] to tune the learning rate.
- Adam is used as the optimizer.
- Language model fine-tuning:
  - We fine-tune the top layer of the forward and backward language models for one cycle of two epochs and then train all layers for one cycle of ten epochs.
- Classifier fine-tuning:
  - We employ gradual unfreezing to prevent catastrophic forgetting:
    - We unfreeze one layer at a time, starting from the last, each time fine-tuning for one cycle of 10 epochs.

Results

Results I

Table: Classes F_1 scores (in %) on the test set.

Class	NB	SVM	F-ULMFiT	B-ULMFiT	F+B-ULMFiT	Count
Casa Civil	72.22	74.29	82.35	83.33	**85.71**	18
Controladoria	**80.00**	**80.00**	**80.00**	0.00	66.67	2
Defensoria Pública	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	8
Poder Executivo	80.00	81.82	78.26	**90.91**	86.96	10
Poder Legislativo	40.00	**100.00**	**100.00**	**100.00**	**100.00**	1
Agricultura	28.57	75.00	**85.71**	75.00	**85.71**	4
Cultura	**91.67**	**91.67**	88.00	**91.67**	**91.67**	13
Desenv. Econômico	**66.67**	**66.67**	28.57	33.33	33.33	4
Desenv. Urbano	75.00	**85.71**	75.00	66.67	**85.71**	4
Economia	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	1
Educação	70.00	**75.00**	72.00	66.67	**75.00**	13
Fazenda	85.71	88.37	86.36	**90.48**	**90.48**	21
Justiça	**80.00**	75.00	66.67	75.00	66.67	5
Obras	84.85	85.71	87.50	87.50	**90.91**	18
Saúde	91.43	93.94	**95.38**	91.43	94.12	32
Segurança Pública	**97.03**	95.24	95.24	96.15	94.34	50
Transporte	91.89	95.00	87.18	**95.24**	**95.24**	20
Meio Ambiente	80.00	**100.00**	66.67	66.67	66.67	2
Tribunal de Contas	**100.00**	**100.00**	95.65	**100.00**	**100.00**	11
Average F	79.74	**87.55**	82.66	79.48	84.69	237
Weighted F1	86.86	89.17	87.46	88.14	**89.74**	237
Accuracy	86.92	89.45	87.76	89.03	**90.30**	237

Results II

- All models performed better than a majority class classifier, which wields F_1 scores of 7.35 and 1.83 and an accuracy of 21.10.
- The SVM and ULMFiT models outperformed the Naive Bayes classifier across almost all categories.
- F_1 scores and accuracies approaching 90.00% indicate good results, though we do not have a human performance benchmark for comparison.

Results III

- SVM and ULMFiT scores are comparable: the former has greater average F_1 score while the latter wins at weighted F_1 score and accuracy.
- SVM has some advantages:
  - Time: SVM takes less than two seconds to train on CPU, while ULMFiT takes more than half an hour on GPU.
  - Simplicity: SVM training is straightforward, while ULMFiT requires three different steps with many parts that need tweaking (gradual unfreezing, learning rate schedule, discriminative fine-tuning).
- Consequence: ULMFiT has more hyperparameters to tune and each search iteration is expensive – the time it takes to train one ULMFiT model is enough to train more than 1,000 SVM models with different configurations of hyper-parameters.

Conclusion

Conclusion

- A new language corpus was presented in the domain of Official Gazettes, for classification of institution of origin.
- We have found that SVM is competitive with ULMFiT, a SOTA technique.
  - In this domain, word order may not be so important and some terms strongly indicate particular classes.

Referências I

Bradbury, J., Merity, S., Xiong, C., and Socher, R. (2016).
Quasi-recurrent neural networks.
CoRR, abs/1611.01576.

Howard, J. and Ruder, S. (2018).
Fine-tuned language models for text classification.
CoRR, abs/1801.06146.

Ioffe, S. and Szegedy, C. (2015).
Batch normalization: Accelerating deep network training by reducing internal covariate shift.
In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, pages 448–456. JMLR.org.

Referências II

Kudo, T. and Richardson, J. (2018).
SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.
In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics (ACL).

Smith, L. N. (2015).
No more pesky learning rate guessing games.
CoRR, abs/1506.01186.

Smith, L. N. and Topin, N. (2017).
Super-convergence: Very fast training of residual networks using large learning rates.
CoRR, abs/1708.07120.

Referências III

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014).
Dropout: A simple way to prevent neural networks from overfitting.
J. Mach. Learn. Res., 15(1):1929–1958.

Inferring the source of official texts: can SVM beat ULMFiT?

Pedro Henrique Luz de Araujo[1] Teófilo Emidio de Campos[1]
Marcelo Magalhães Silva de Sousa[2]

[1]Department of Computer Science, University of Brasília, Brasília – DF, Brazil

[2]Tribunal de Contas do Distrito Federal, Zona Cívico-Administrativa, Brasília – DF, Brazil

teodecampos@unb.br

Sponsored by FAPDF and CNPq

PROPOR 2020

Paper and code available at
`https://cic.unb.br/~teodecampos/KnEDLe/propor2020/`