University of Brasilia

Institute of Exact Sciences
Department of Computer Science

From Documents to Entities:

A journey through Natural Language Processing tasks and domains

Pedro Henrique Luz de Araujo

Document presented for examination of the Master Degree in Computer Science

Supervisor
Dr. Teófilo E. de Campos

Brasilia
2020

University of Brasilia

Institute of Exact Sciences
Department of Computer Science

From Documents to Entities:

A journey through Natural Language Processing tasks and domains

Pedro Henrique Luz de Araujo

Document presented for examination of the Master Degree in Computer Science

Dr. Teófilo E. de Campos (Supervisor)
CIC/UnB

Dr. Alexandre Rademaker	Dr. Thiago Faleiros
IBM Research	CIC/UnB

Dr. Genaina Nunes Rodrigues
Computer Science Graduate Program Coordinator

Brasilia, 23 June 2020

Abstract

Every day a massive amount of data is produced—a significant part of it in natural language text ranging from various domains (social media posts, books, news, official reports, legal proceedings). This rich source of information can produce usable knowledge. The challenge is that natural language texts are unstructured: processing is required to obtain insight and structured knowledge from the data.

Natural Language Processing (NLP) has seen a great deal of progress in the last decade, but current models require a large number of annotated examples and tend to not generalise beyond training data and domain. Though recent transfer learning approaches mitigate those needs, specific-domain labelled datasets are still needed to fine-tune pre-trained models and for evaluation. In this work we study NLP tasks across different domains, developing datasets in the legal and public administration domains, exploring techniques for low-resource areas of application and showing experimental results that evaluate Bag-Of-Words (BOW) and Deep Neural Networks (DNN) models trained on the data.

We perform five sets of experiments across different tasks, datasets and domains: 1) We propose and examine a dataset for classification of legal documents, comparing different models and approaches; 2) We propose a dataset of Official Gazette texts with labelled and unlabelled data and use it to compare traditional BOW models to a SOTA transfer learning method, where we find the former to be surprisingly competitive; 3) We employ Latent Dirichlet Allocation (LDA) to discover topics present in our dataset of legal documents and explore their use as a text representation method for classification; 4) We propose a dataset for Named Entity Recognition (NER) in legal documents with domain specific entities; 5) We introduce and plan a final study on Entity Linking (EL).

Keywords: natural language processing, text classification, topic models, named entity recognition, entity linking, transfer learning

Resumo

Todos os dias uma quantidade massiva de dados é produzida—grande parte em textos de variados domínios (*posts* de redes sociais, livros, notícias, relatórios oficiais, processos jurídicos). Dessa rica fonte de informação pode-se obter conhecimento utilizável. No entanto, sua natureza não-estruturada exige processamento para se obter *insights* e conhecimento estruturado.

O Processamento de Linguagem Natural (PLN) progrediu muito na última década, mas modelos atuais precisam de muitos exemplos anotados e tendem a não generalizar além dos dados e domínio de treinamento. Embora abordagens de transferência de aprendizado recentes tenham mitigado isso, conjuntos de dados rotulados de domínio específico ainda são necessários para ajuste fino de modelos pré-treinados e para avaliação. Nesse trabalho, estudamos tarefas de PLN em diferentes domínios, desenvolvendo conjuntos de dados de textos jurídicos e da administração pública, explorando técnicas para áreas de aplicação com poucos recursos e exibindo resultados experimentais que avaliam modelos de saco-de-palavras e de redes neurais profundas treinados nos dados.

Realizamos cinco conjuntos de experimentos em diferentes tarefas, conjuntos de dados e domínios: 1) Propomos e examinamos um conjunto de dados para classificação de documentos jurídicos, comparando diferentes modelos e abordagens; 2) Propomos um conjunto de textos de Diário Oficial, com dados anotados e não anotados, e usamo-lo para comparar modelos de saco-de-palavras com uma técnica estado-da-arte de transferência de aprendizado, concluindo que aqueles são surpreendentemente competitivos; 3) Usamos Alocação de Dirichlet Latente para descobrir tópicos presentes no conjunto de documentos jurídicos e exploramos seu uso como uma forma de representação de textos para classificação; 4) Propomos um conjunto de documentos jurídicos para Reconhecimento de Entidade Nominada com entidades específicas do domínio; 5) introduzimos e planejamos um estudo final sobre Ligação de Entidade.

Palavras-chave: processamento de linguagem natural, classificação de texto, modelos de tópicos, reconhecimento de entidade nomeada, ligação de entidade, transferência de aprendizado

Contents

List of Acronyms and Abbreviations

ARE *Agravo de Recurso Extraordinário*

BERT Bidirectional Encoder Representations from Transformers

BOW Bag-Of-Words

CG Candidate Generation

CNN Convolutional Neural Network

CRF Conditional Random Fields

DNN Deep Neural Networks

ED Entity Disambiguation

EL Entity Linking

FC Fully-Connected

GPT-2 Generative Pre-trained Transformer 2

KB Knowledge Base

LDA Latent Dirichlet Allocation

LSI Latent Semantic Indexing

LSTM Long Short-Term Memory

MD Mention Detection

ML Machine Learning

NB Naïve Bayes

NER Named Entity Recognition

NLP Natural Language Processing

OCR Optical Character Recognition

PLSI Probabilistic Latent Semantic Indexing

QRNN Quasi-Recurrent Neural Network

RE *Recurso Extraordinário*

ReLU Rectified Linear Unit

RNN Recurrent Neural Networl

SGD Stochastic Gradient Descent

SOTA State-of-the-art

STF *Supremo Tribunal Federal*

SVD Singular Value Decomposition

SVM Support Vector Machines

tf-idf term frequency-inverse document frequency

ULMFiT Universal Language Model Fine-tuning

XGBoost eXtreme Gradient Boosting

Notation

Dir	The Dirichlet distribution
$\mathcal{E}$	Entity Set
Multinomial	The Multinomial distribution
ReLU	The ReLU function
$\mathcal{V}$	Vocabulary size

Bold lower case letters are used to represent vectors ($\mathbf{x}$), while bold upper case letters are employed to indicate matrices ($\mathbf{X}$) and italic lower case letters are used for scalars (x). Italic upper case letters are used to denote both sets and sequences (X).

Chapter 1

Introduction

Modern human society constantly produces data—a significant part of it in natural language text ranging from various domains: social media posts, books, news, official reports, legal proceedings. The challenge is that this rich source of information is unstructured and requires processing in order to produce useful knowledge. Humans are no strangers to this task: legal workers read case files in order to categorize them; researchers analise medical files to find relations between populations and health issues; auditors examine documents to search for frauds and irregularities. But human labour, though (reasonably) accurate, is expensive and slow. Machines can come at our aid: Natural Language Processing (NLP) techniques enable computers to analyse and structure text data, freeing time that humans can use to perform more complex, creative tasks.

NLP has seen a great deal of progress in the last decade. This has been in great part to the use of deep neural network architectures, which have pushed the state of the art of tasks like sentiment analysis [1, 2, 3], machine translation [4, 5, 6] and natural language inference [7, 1, 8]. Unfortunately, in addition to requiring a large number of annotated examples, deep NLP models tend to not generalise beyond training data and domain [9]. A named entity recogniser trained on a news corpus will not perform as well when applied to legal documents, for example.

Transfer learning can help by reducing the amount of labelled target data needed to achieve good results. Using word embeddings [10, 11, 12] pre-trained on large corpora is a transfer learning method that has become pervasive in the NLP field. More recently, efforts have turned to pre-training language models [13, 2, 3, 14], as these provide contextualized embeddings that greatly improve language representation—instead of one fixed vector for each word the embedding will depend on local context and disambiguate homonyms (e.g. different embeddings for the weapon *bow* and the gesture *bow*). That said, having labelled datasets for specialized domains is still necessary; be it for fine-tuning or for evaluation.

In this work we explore NLP tasks across different domains, following a top-down approach: first we examine documents as a whole; then, we focus on entities mentioned in documents. In order to promote research on under-explored domains, we help developing new datasets with legal and public administration texts. We also train shallow and deep learning models on these datasets to establish benchmarks for comparison.

1.1 Objectives

This dissertation aims to examine how NLP can be used to process documents from specific domains by performing experiments across different tasks and datasets. More specifically, we aim to:

- propose datasets for classification of legal (§ 2.1) and public administration (§ 2.2) texts, and compare models trained on them;
- train topic models on legal texts, analyse topic semantics and experiment with topic distribution as text representation for classification (Chapter 3);
- propose a dataset for Named Entity Recognition (NER) on legal data with specific classes for legal entities, and train a model on the data (Chapter 4);
- propose an EL system for low-resource domains (Chapter 5).

1.2 Contributions

The present work has resulted in the following contributions:

- VICTOR, a dataset of legal documents from Brazil's Supreme Court labelled by experts and an evaluation of models trained on it;
- a dataset of labelled and unlabelled Official Gazette documents, a comparison between traditional shallow models and a state-of-the-art approach that uses deep transfer learning, and an ablation analysis of the latter;
- a topic analysis of Brazil's Supreme Court documents and an empirical assessment of topic distribution as text vectorization;
- LeNER-Br, a dataset of legal documents for NER and a deep learning model trained on it.

The work has generated the following publications:

- Luz de Araujo, P. H. et al. VICTOR: a dataset for Brazilian legal documents classification. [15]
- Luz de Araujo, P. H. et al. Inferring the source of official texts: can SVM beat ULMFiT? [16].
- Luz de Araujo, P. H. et al. LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text [17]

1.3 Outline

We divide the dissertation in two parts: i) document-level applications and ii) entity-level applications.

Chapter 2 - Text Classification: introduces the Document-level Part; we present two text classification tasks in the legal and public administration domains.

Chapter 3 - Topic Modelling: we describe how we used Latent Dirichlet Allocation (LDA) to model legal documents, analysed topic semantics, and examined topic distribution as text representation.

Chapter 4 - Named Entity Recognition: introduces the Entity-level Part; we propose a NER dataset of manually annotated legal texts, describe how we trained a entity recogniser on it, and analyse the obtained results.

Chapter 5 - Proposal for Entity Linking: we propose an investigation of EL for low-resource domains and detail the next steps.

Part I

Document-level

Chapter 2

Text Classification

Text classification is a Natural Language Processing (NLP) task concerned with assigning one or more classes or categories to a contiguous sequence of words, such as a sentence, a paragraph or a document. Text classification research includes building datasets, designing features or developing classifiers [18]. Applications include spam filtering [19], sentiment analysis [20] and topic identification [21]. Since it is often the case that some terms strongly indicate specific classes, word order is not so important for many text classification tasks[1]. Given that, Bag-Of-Words (BOW) models with term frequency-inverse document frequency (tf-idf) features of unigrams or bigrams usually achieve good performance and serve as strong baselines [22].

The task may be described as follows. Given a corpus of n documents (or sentences, paragraphs, tweets etc), $D = \{d_1, d_2, \cdots, d_n\}$, and a set of k classes, $C = \{c_1, c_2, \ldots, c_k\}$, text classification aims to assign to each document in D one or more of the classes in C. Single-label classification problems include binary (spam or not spam) and multi-class (positive, negative or neutral sentiment) problems, where each document must be assigned to only one class. On the other hand, in multi-label problems each document can be assigned to more than one category.

In this chapter we present two cases of text classification problems. Both of them are supervised classification tasks, but they each deal with different corpora with particular characteristics and specific domains.

In Section 2.1, we propose a dataset of Brazilian lawsuits with two classification tasks, one single-label and the other multi-label. We also establish benchmarks for each task using different methods for text representation and classifiers. We show that BOW models with tf-idf features are a strong choice for classifying long texts.

[1] One important exception to this rule is sentiment analysis tasks, where a “not” before a “good” dramatically alters the sentiment polarity.

In Section 2.2, we introduce a dataset of documents from the Official Gazette of the Federal District, containing both unlabelled samples and samples annotated with their public entity of origin. We compare BOW models to a transfer learning approach using Universal Language Model Fine-tuning (ULMFiT), finding that SVM is surprisingly competitive.

2.1 VICTOR: a dataset for Brazilian legal documents classification

This section describes VICTOR, a novel dataset built from Brazil's Supreme Court digitalized legal documents, composed of more than 40 thousand appeals, which includes roughly 692 thousand documents—about 4.6 million pages. The dataset contains labelled text data and supports two types of tasks: document type classification; and theme assignment, a multi-label problem. We present baseline results using bag-of-words models, convolutional neural networks, recurrent neural networks and boosting algorithms. We also experiment using linear-chain Conditional Random Fields (CRF) to leverage the sequential nature of the lawsuits, which we find to lead to improvements on document type classification. Finally we compare a theme classification approach where we use domain knowledge to filter out the less informative document pages to the default one where we use all pages. Contrary to the Court experts' expectations, we find that using all available data is the better method[2].

2.1.1 Introduction

Brazil's legal system suffers from an unreasonably large number of lawsuits [23]. To put matters into perspective, about 80 million lawsuits were awaiting judgement in 2017. That is almost one process for every three Brazilians. The period from 2009 to 2017 saw an increase of 19.4 million lawsuits [24]. In addition, the average processing time of lawsuits can reach more than seven years in some cases. The long waiting times impact Brazil's legal certainty and represent greater budgetary requirements—Brazil spent R$ 90.7 billion in 2017 to maintain the judiciary, approximately 28 billion[3] dollars [25].

This section describes an effort to apply Natural Language Processing (NLP) and Machine Learning (ML) techniques to Brazil's Supreme Court—*Supremo Tribunal Federal* (STF)—cases to help overturn this scenario. The STF receives roughly 42 thousand cases

[2] An early version of this section has been published in: Luz de Araujo, P. H. et al. VICTOR: a dataset for Brazilian legal documents classification [15].

[3] Considering average exchange rate of 2017: 3.19 reais to 1 dollar.

each semester, taking 22 thousand hours for humans to sort through. That time could be better spent at more complex stages of the judicial work flow, for instance the ones requiring legal reasoning.

Most of the cases reach the court as PDF files with raster scanned documents. Approximately 10% of these are unstructured, containing several unindexed documents ranging from petitions and orders to rulings. Therefore, as a first goal we explore and evaluate methods for automatically classifying document types. The documents originate in different Brazilian courts and often contain visual noise (handwritten annotations, stamps, stains). So the main challenges here are the intra-class diversity and the quality of the scanned documents.

In addition, lawsuits pertaining to the STF belong to one or more general repercussion (*repercussão geral*) themes that are presently checked by humans during the initial processing of the suit. As our final goal we train and evaluate a series of models that assign themes to suits. In this case, the central difficulty is the size of the suits, which can contain dozens of documents.

This section's main contribution is VICTOR, a dataset of legal documents belonging to STF's suits labelled by a team of experts. We hope that this can help other researchers to explore NLP and ML applied to the legal field, document analysis, text classification and multi-label classification. The second contribution is a benchmark that compares a series of models we evaluate for each goal: document type classification and lawsuit theme assignment.

The rest of this section is organised as follows. We first introduce other works related to text classification and processing of legal domain documents (2.1.2). Then we discuss the dataset and its creation process (2.1.3). We present the models explored and the experiments involved and discuss the results obtained regarding the first (2.1.4) and second goals (2.1.5), respectively. Finally, we conclude the work by presenting our final considerations (2.1.6).

2.1.2 Related work

Text classification

A traditional well-performing baseline for text classification is representing a document as a Bag-Of-Words (BOW) and give that as input to a classifier like Naïve Bayes (NB) or Support Vector Machines (SVM) [26]. This representation is invariant to word-order, a property that may hinder performance in applications such as sentiment classification, where word positioning can completely change the semantics of the sentence. Using n-grams instead of only 1-grams (words) can mitigate that problem. Joulin et al. [27]

propose a shallow model that uses n-gram features and hierarchical softmax to efficiently train on large datasets. Liu et al. [28] propose a semi-supervised text classification method that combines boosting and examples that do not belong to any class, which is shown to particularly benefit problems with few labelled examples.

The popularization of deep neural networks gave rise to the creation of many architectures for text categorization. Zhang et al. [18] and Conneau et al. [29] independently show that a character-level CNN surpasses shallow models' performances on large datasets. Johnson and Zhang [30] were able to improve the state of the art by using a word-level LSTM network with pooling. Howard and Ruder [31] introduce a task-agnostic transfer learning method that outperforms the state-of-the-art text classifiers, in addition to requiring much less data to match the performance of a model trained from scratch.

NLP and ML in the legal domain

Several works have explored the use of NLP and ML techniques to analyse legal documents. Named Entity Recognition (NER) has been used to automatically extract relevant entities from legal text [32, 33, 17]. Automatic summarization has been employed to help manage the great amount of information legal employees are required to process [34, 35, 36, 37]. In addition, topic models have been used to analyse large corpora of legal documents [38, 39, 40].

Text classification in the legal domain is used in a number of different applications. Katz et al. [41] use extremely randomized trees and extensive feature engineering to predict if a decision by the Supreme Court of the United State would be affirmed or reversed, achieving an accuracy of 69.7%. Aletras et al. [42], in a similar fashion, trained a model to predict, given the textual content of a case from the European Court of Human Rights, if there has been a violation of human rights or not. The paper employed n-grams and topics as inputs to an SVM, reaching an accuracy of 79%. Şulea et al. [43] trained a linear SVM on text descriptions of cases from the French Supreme Court, obtaining a 90% F_1 score in law area prediction (eight classes) and a 96.9% F_1 score in ruling prediction (six classes). Undavia et al. [44] evaluated a series of classifiers (CNN, RNN, SVM and logistic regression) trained on a dataset of cases from the American Supreme Court. Their best performing model, a Convolutional Neural Network, was able to achieve an accuracy of 72.4% when classifying the cases into 15 broad categories and 31.9% when classifying over 279 finer-grained classes.

2.1.3 The dataset

The VICTOR[4] dataset is composed of 45,532 Extraordinary Appeals[5] (*Recursos Extraordinários*) from the STF. Each suit in turn contains several different documents, ranging from the appeal itself to certificates and rulings, adding up to 692,966 documents comprising 4,603,784 pages.

The Court provided the VICTOR data in the form of PDF files where each file either represents a particular document or is an unstructured volume containing several documents. In the former case, the suits were manually annotated by experts from the Court staff with labels for the document classes, amounting to 44,855 suits with 628,820 documents.

The first issue we faced was extracting the text from the PDF files. A significant part of the provided data is was available as images scanned from printed documents, which often contain handwritten annotations, stamps, stains and other sources of visual noise.

The first step was checking if a file content was purely an image scan or contained text data. If the former was true, the pipeline applied an Optical Character Recognition (OCR) system [45] and stored the resulting text. Otherwise, regular expressions were used to verify the embedded text quality. In case the quality is deemed acceptable, the text was stored; if not, OCR was appleid and its result stored. The extracted text contained some artifacts from the OCR system and PDF tagging scheme. For that reason, the pipeline employed regular expressions to clean the text. In addition, some preprocessing steps were applied: stemming, removal of stop words, lower-casing, tokenization of e-mails and URLs, and specific tokenization of articles of law (*Lei*—law—11.419 to LEI_11419)[6].

The data contains two types of annotation for two different tasks.

1. Labels for document type classification: *Acórdão*, for lower court decisions under review; *Recurso Extraordinário* (RE), for appeal petitions; *Agravo de Recurso Extraordinário* (ARE), for motions against the appeal petition; *Despacho*, for court orders; *Sentença* for judgments; and *Others* for documents not included in the previous classes. This task has evolved from early versions evaluated in [46, 47].

2. Labels for lawsuit theme classification, which assign one or more General Repercussion (*Repercussão Geral*) themes to each Extraordinary Appeal. There are 28 theme options identified by integers (e.g. theme 810) corresponding to the most relevant ones, which were chosen by the Court workers, and one class (with ID 0) for the remaining themes, summing up to 29 classes.

[4] The project name is a tribute to the late Justice Victor Nunes Leal.

[5] Appeals on the grounds of conflit with Consitutional Law.

[6] The preprocessing pipeline—from text extraction to tokenizing—was developed and executed by other members of the Victor Project.

To ensure the reproducibility of our experiments we randomly divided the appeals into 70%/15%/15% splits for train/validation/test respectively, maintaining theme distribution across them.

There are three versions of VICTOR:

- Big VICTOR or BVic, used only for theme classifications, since it contains all data (45,532 suits), including the unlabelled documents (677 suits).
- Medium VICTOR or MVic (44,855 suits, 628,820 documents and 2,086,899 pages) is the result of filtering out unlabelled samples and can be employed for both theme and document type classification.
- Small VICTOR or SVic. Due to the huge size of the MVic dataset, it is extremely hard to share it (text data and source image data) with the community. So we limit the number of suits for each theme to 100 samples in each set to create the SVic dataset, which contains 6,510 Extraordinary Appeals, 94,267 documents and 339,478 pages.

Table 2.1 exhibits the document type distribution for each split of the relevant versions of the dataset. Figures 2.1, 2.2 and 2.3 show the theme distribution for each versions of VICTOR. The presented theme IDs are the ones originally used by the Court[7].

Table 2.1: Document type distribution per split.

Dataset	Category	Training set		Validation set		Test set	
		Documents	Pages	Documents	Pages	Documents	Pages
MVic	Acórdão	1,966	4,740	354	656	358	659
	ARE	2,894	34,640	760	8,373	721	7,347
	Despacho	2,415	3,952	326	457	346	490
	Others	420,494	1,323,841	92,696	280,399	93,855	283,763
	RE	4,396	77,893	902	15,753	849	15,129
	Sentença	4,065	21,210	727	3,970	696	3,627
SVic	Acórdão	301	553	201	299	199	273
	ARE	270	2,546	237	2,149	213	1,841
	Despacho	265	346	147	183	147	198
	Others	38,585	134,134	25,898	84,104	25,744	85,408
	RE	453	9,509	326	6,364	312	6,331
	Sentença	420	2,129	284	1,636	265	1,475

[7] A list of all themes is available at `http://www.stf.jus.br/portal/jurisprudenciaRepercussao/abrirTemasComRG.asp`.

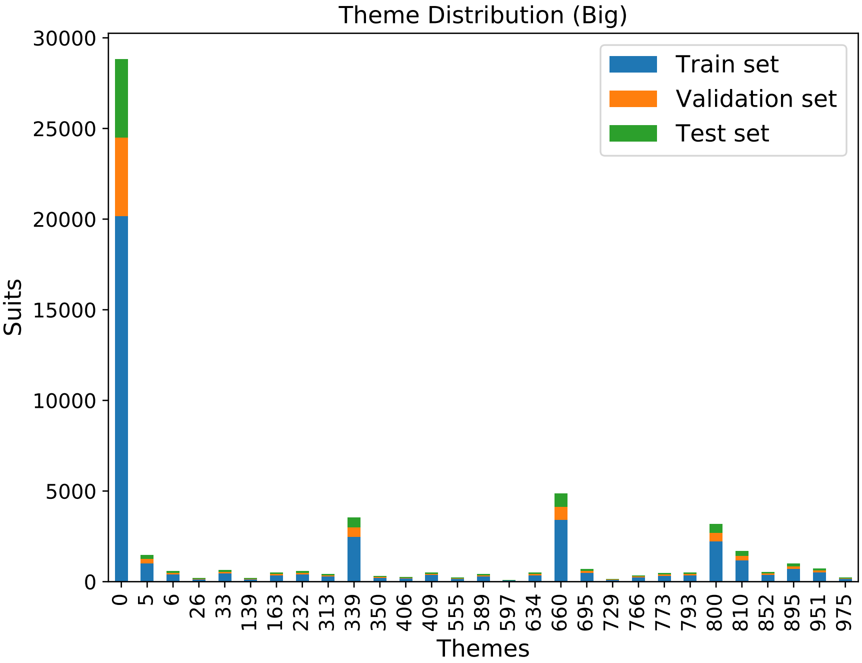


Figure 2.1: BVic theme distribution.

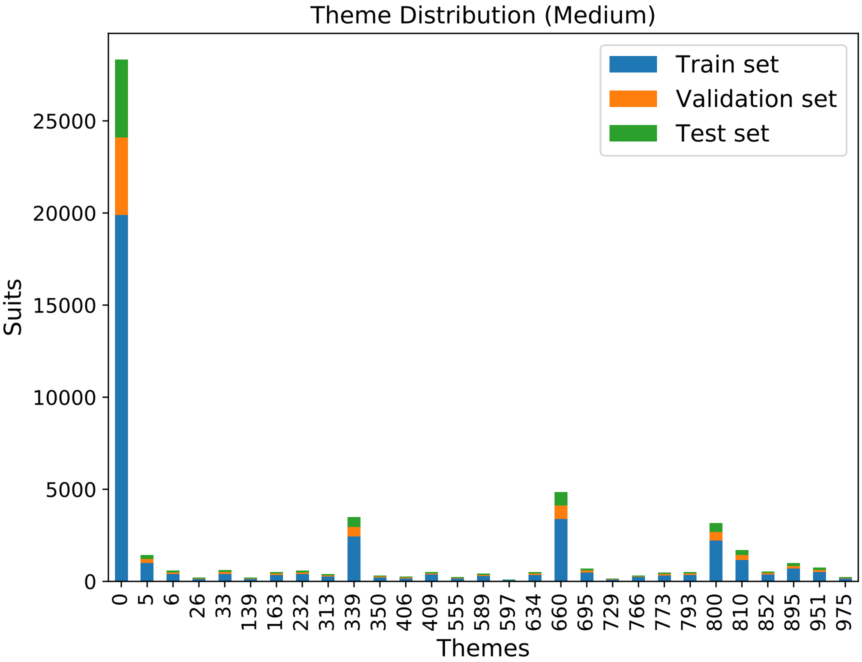


Figure 2.2: MVic theme distribution.

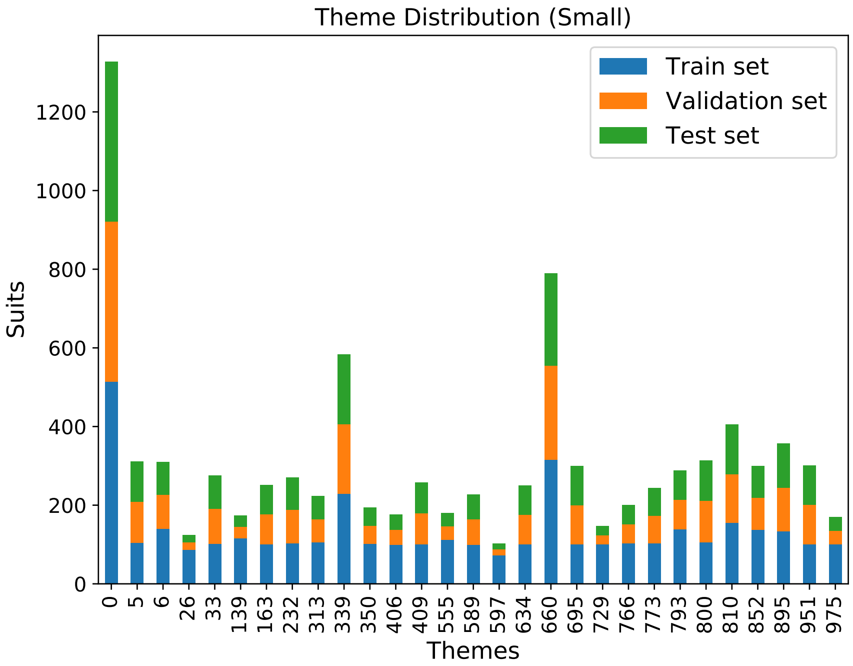


Figure 2.3: SVic theme distribution.

2.1.4 Document type classification

Here we compare the different methods explored to classify the document types. All results, unless stated otherwise, are reported on the test set and refer to page prediction accuracy. For a baseline, we select the most frequent class (*others*), which gives, on M/SVic test set, an F_1 score weighted by class frequencies of 87.06/84.41 and an average F_1 score of 15.90/15.73. We run experiments with two BOW methods and two deep DNN architectures.

BOW methods

We represent each document as a bag-of-words with tf-idf features. We experiment with two different classifiers: Naïve Bayes and SVM.

Feature extraction: We search for the best hyperparameters using the validation set. The best approach uses unigrams and bigrams, and includes only terms with a minimum document frequency of two pages and a maximum frequency of 50% of the pages. We restrict our vocabulary to the 70,000 most frequent words in the training set.

NB: We train a Naïve Bayes classifier with an additive Laplace smoothing parameter $\alpha = 0.001$ and class prior fitting due to the category imbalance.

SVM: We employ an SVM with linear kernel and apply weights inversely proportional to class frequencies to compensate the imbalance.

Convolutional Neural Network

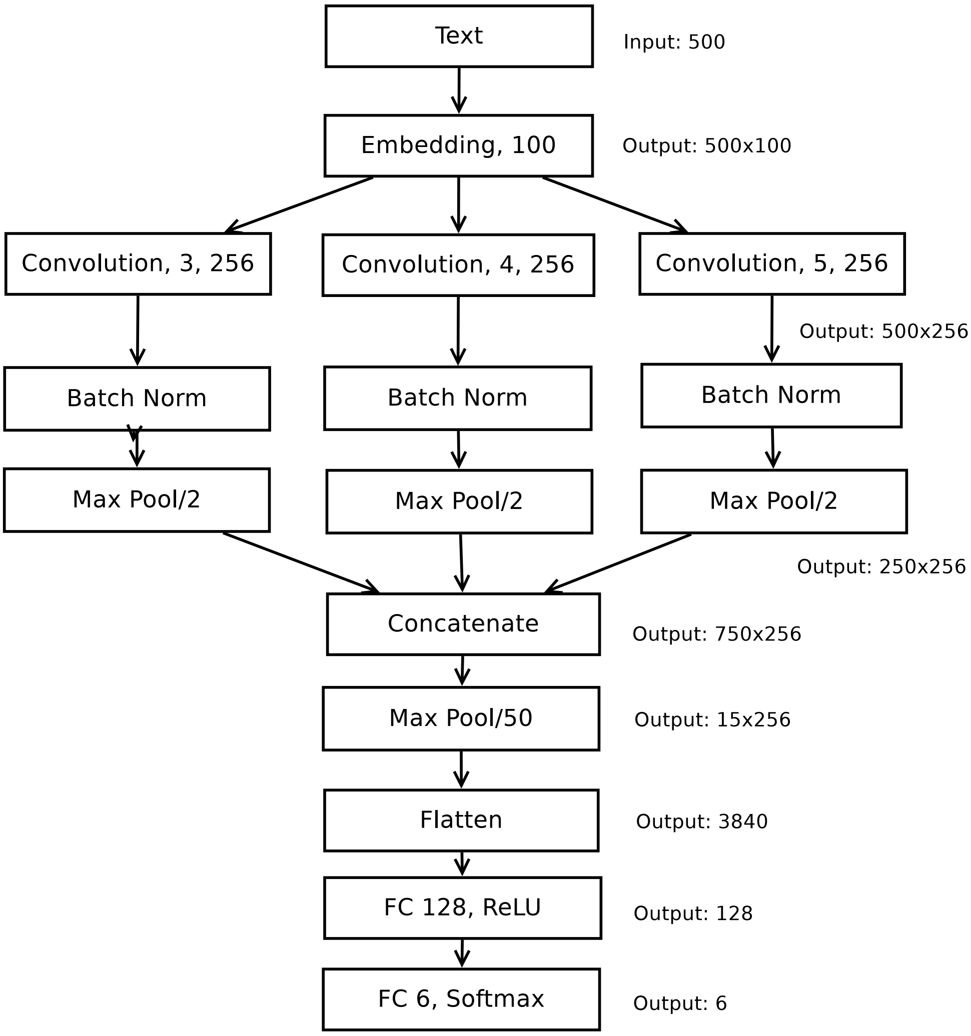


Figure 2.4: CNN architecture for document type classification.

We based our CNN architecture on the one proposed in [29]. Our network is shallower though, as stripping several layers improved the accuracy of the model. As a result, the network trains faster and requires less GPU memory. We also work on the word level instead of on the character level.

The architecture is shown in Figure 2.4. The network takes as input the first 500 tokens from the input and embed them into 100 dimensional vectors. The remaining tokens are discarded, with the intuition that those first tokens are sufficient to discriminate between classes. Next, we concatenate the output of three convolutional blocks formed by a convolutional layer with 256 filters and varied sizes (3, 4 and 5) followed by batch normalization and max pooling layer of size 2. Another max pooling operation (of size 50) is applied to the result of the concatenation and the output is flattened. Finally, the flattened tensor is processed by two fully connected layers and a softmax function produces the final output. A dropout mask is applied to the first fully connected layer with 50% dropping probability.

We use Adam [48] to optmise the cross-entropy loss function with a learning rate of 0.001 and train the model for 20 epochs with mini-batches of 64 samples.

Bidirectional LSTM Network

For this model, we embed the first 500 tokens from each page into an 100 dimensional space—like we did for the CNN—and subsequently feed them into a Bidirectional [49] Long Short-Term Memory (LSTM) [50] layer with 200 units for each direction. The forward and backward representations of the sequence are summed together and fed to a fully connected layer followed by a softmax activation that calculates the final class probabilities. Figure 2.5 exhibits the architecture.

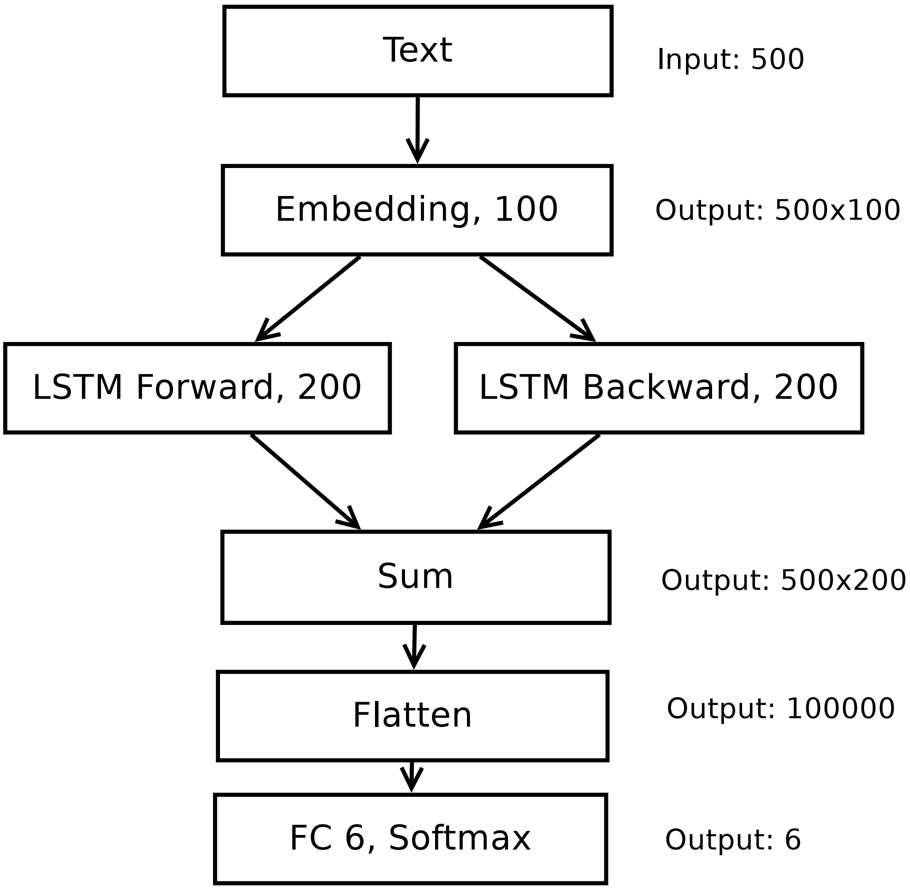


Figure 2.5: Bi-LSTM architecture for document type classification.

We trained the model for 20 epochs with batches of 64 samples and learning rate value of 0.001 with Adam optmiser.

Linear-chain CRF post-processing

Instead of classifying each page by itself, one can use the fact that a lawsuit is composed by a series of document pages and treat the document classification as a sequence labelling problem. Intuitively, a page is more likely to be followed by another of the same type, as documents usually contain more than one page, so taking in consideration the sequential aspect of the data should improve classification metrics.

Rather than having a page as input and outputting a document type prediction, the sequence labelling approach outputs a series of type predictions (tags) given a series of input pages. We can consider neighbor tag information by employing linear-chain CRF, which have been shown to be very effective in sequence tagging problems [51, 52, 53].

To better leverage the sequential information, we adapt the document classes by using the IOB tagging scheme [54]. We prepend "B-" to the ground truth of first pages of document or "I-" in the other cases (e.g. if a suit begins with a RE of three pages, the sequence of labels would start with B-RE, I-RE, I-RE). The training instances are the dataset suits, which are sequences of pages. We pre-calculate a six-dimensional embedding for each page by feeding it to our best performing model, the CNN, and saving the output of the softmax. The sequences of page embeddings are then used to train a CRF model.

We employ said procedure in both MVic and SVic. The following section compares the performance of the CNN model before and after the CRF processing for each test set.

Results and discussion

Table 2.2 compares test performance across the evaluated models. The CNN and the BiLSTM trained and evaluated on MVic outperform the other models in all categories; the SVM followed close behind, while the NB classifier achieved much lower scores. Furthermore, all models are able to beat the baselines for weighted and average F_1 score, with the exception of the NB, whose weighted F_1 is 2.63 p.p. lower, though the average F_1 score is much higher than the baseline. The CNN result represents a relative increase of 8.71% and 344.00%, respectively, for each metric. We can see that, due to the imbalanced nature of the data, the average F_1 is a more informative metric of the performance of the model.

Regarding the SVic dataset, the SVM and the CNN were the best-performing models. Similarly to the MVic scenario, all models beat the baseline, with the CNN representing a relative increase of 12.22% and 381.99% for the weighted and average F_1 score, respectively. These results suggest that the SVM is able to better generalise the much smaller dataset.

Table 2.2: F_1 score (in %) of our methods for document type classification on the test sets. A baseline that always chooses the majority class yields an F_1 score weighted by class frequencies of 87.06/84.41 and an average F_1 score of 15.90/15.73 on MVic and SVic, respectively.

Dataset	Model	Acórdão	ARE	Despacho	Others	RE	Sentença	Weighted	Average
MVic	NB	49.20	32.08	39.82	89.38	38.06	37.80	84.77	47.72
	SVM	65.41	52.62	59.34	95.85	64.52	69.75	92.88	67.92
	BiLSTM	**72.84**	57.82	**60.07**	97.11	67.74	69.96	94.33	**70.92**
	CNN	71.06	**58.11**	56.04	**97.37**	**68.71**	**72.35**	**94.64**	70.61
SVic	NB	66.40	36.07	51.15	93.24	55.89	55.99	88.93	59.79
	SVM	81.15	**58.06**	**67.88**	96.85	74.66	**79.30**	94.25	**76.32**
	BiLSTM	85.82	52.12	51.01	97.15	74.06	76.70	94.65	72.81
	CNN	**86.43**	55.92	59.88	**97.30**	**76.23**	79.29	**94.72**	75.84

In both scenarios and across all explored models, the category *Others* has the best F_1 score. This is not surprising, since it includes the vast majority of pages in the datasets. That being said, our strategies for dealing with data imbalance were effective—without fitting the class prior (NB) or using class weights (SVM, CNN, and BiLSTM) the classifiers behaved approximately as the baseline, predicting almost every sample as belonging to the *Others* class.

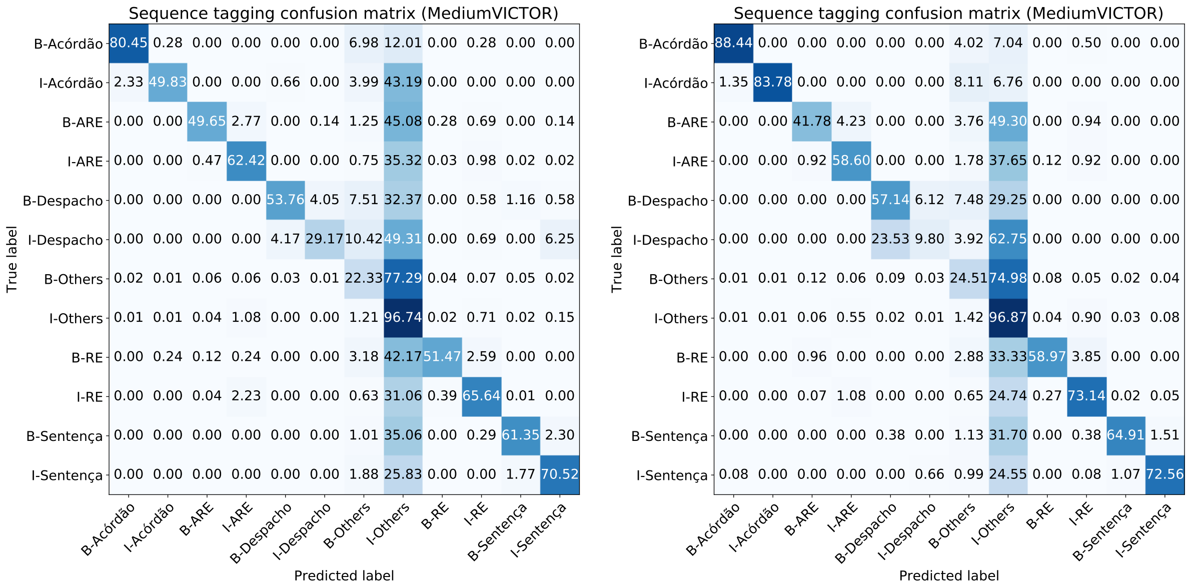
Table 2.3 shows the impact of CRF modeling. Our sequence modeling approach, albeit simple, results in overall improvements in both versions of dataset. The best increase in performance was regarding *Despacho* classification on MVic—a relative improvement of 11.62%. On the other hand, SVic's *Despacho* saw a relative decrease of 5.33%. The MVic model had the greatest positive changes, perhaps due to the fact that the MVic CNN model had more room for growth than its small counterpart and more training data.

Table 2.3: F_1 scores (in %) before and after CRF processing on the test sets.

	MVic		SVic	
Classes	CNN	CNN-CRF	CNN	CNN-CRF
Acórd.	71.06	75.02 / +5.57%	86.43	90.60 / +4.82%
ARE	58.11	62.89 / +8.23%	55.92	59.54 / +6.47%
Desp.	56.04	62.55 / +11.62%	59.88	56.69 / -5.33%
Others	97.37	97.66 / +0.30%	97.30	97.68 / +0.39%
RE	68.71	74.38 / +8.25%	76.23	78.77 / +3.33%
Sent.	72.35	77.77 / +7.49%	79.29	81.13 / +2.32%
Wtd.	94.64	95.37 / +0.77%	94.72	95.33 / +0.64%
Avg.	70.61	75.05 / +6.29%	75.84	77.40 / +2.06%

Figure 2.6 exhibits the confusion matrices of CRF tag predictions. The greatest source of confusion is the I-Others tag (pages classified as others that are not the first page of a document), which is not surprising due to its overabundance. We have a similar scenario

when we analyse the confusion between predictions before and after CRF processing (Figure 2.7): the CRF is more likely to tag a page as *Others* when compared to the original model.



(a) MVic.

(b) SVic.

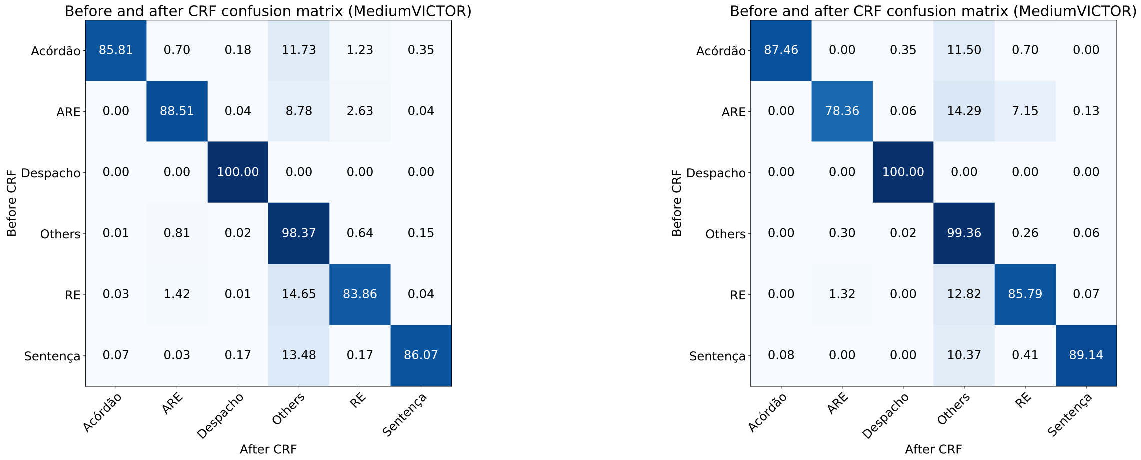
Figure 2.6: Confusion matrix of CRF predictions for the test set and ground truth tags. Each value represents the percentage of samples from the row class that were classified as being from the column class.

One possible way to improve the sequence tagging approach is leveraging the sequential information during the document embedding step, that is, using an end-to-end approach where we jointly train the CRF layer and the feature extractor. Furthermore, our technique employs a vector of 6 dimensions that, while sufficient for our viability assessment needs, cannot sufficiently encode relevant document attributes. Higher dimensional embeddings should improve the task accuracy.

2.1.5 Lawsuit theme classification

BOW Methods

For the task of lawsuit theme classification we represent each document as a vector of tf-idf features. This approach is better suited than using CNNs or RNNs due to the great size of the samples, where dozens—or even hundreds—of pages are not uncommon. Besides the classifiers we mentioned in the previous section, we also train an eXtreme Gradient



(a) MVic.

(b) SVic.

Figure 2.7: Confusion matrix of test set predictions before and after CRF processing. Each value represents the percentage of samples with the row class prediction before CRF processing that were classified as being from the column class after CRF processing.

Boosting (XGBoost) [55] classifier. XGBoost is an optmised tree boosting system that has become very popular amongst Kaggle [8] competitions for various ML tasks.

Since theme classification is a multi-label and multi-class problem we employ an one-vs.-rest approach where we train one classifier for each class and set a threshold value for assigning a theme to a document. That is, given C the set of all possible classes, t the threshold value, $f_c(\cdot)$ the classifier's function for class c, and a document d:

$$\forall c \in C,\ \text{we assign } c \text{ to } d \text{ if } f_c(d) \geq t\,. \tag{2.1}$$

We use 0.5 as the threshold value. All the following reported metrics are on the test set. As a baseline result we choose to assign all themes to all documents, which gives us an F_1 score weighted by class frequencies of 41.17 /40.87/10.87 and an average F_1 score of 5.48/5.49/6.52 on B/M/SVic test set.

Feature extraction: The best performing configuration on the validation set uses only unigrams with a minimum document frequency of 10%. We also limit the vocabulary to the 10,000 most frequent words.

NB and SVM: We employ the same hyperparameters discussed in 2.1.4.

XGBoost: We train 500 trees with a maximum depth of 4 and a shrinkage factor of 0.1.

[8] Kaggle (`https://www.kaggle.com/`) is a online community of data scientists and machine learning practitioners that hosts competitions and offers a cloud-based workbench with GPU support.

Theme classification with domain Knowledge

An intuition legal experts have is that the most informative pages about a suit's themes are the ones not classified as *Others*. On that premise, one possible improvement for theme classification models is to take into consideration only the suit's pages that do not have an *Others* label.

On the other hand, at test time we do not have ground truth knowledge about page type classification. Thus, such method can propagate errors from the document type classification model, which may negatively impact accuracy. To test the feasibility of the idea, we train and test an XGBoost model only with the relevant pages of BVic to establish an upper-bound of performance. When we eliminate all pages labelled as *Others* we lose the suits that contain no other kinds of pages. To establish a fair comparison to a method that uses no domain knowledge, we also train a model on the same suits without removing pages labelled as *Others*. We show the results in the next section.

Results and discussion

Table 2.4 exhibits the models' performance in each VICTOR version. All models are able to beat the baselines for both weighted and average F_1 score. The XGBoost outperforms the other models across all versions of VICTOR, excluding a few themes better assigned by the SVM, and, on two occasions, the NB. Furthermore, the SVM overall results were fairly consistent through the different datasets in comparison with the NB and the XGBoost.

The data imbalance impact on the results here is far less pronounced than in the previous task. XGBoost, the best classifier, has very similar weighted and average F_1 scores in all versions of VICTOR, even though the theme distribution is heavily skewed towards class 0. In addition, the model greatly outperforms the baselines in both averaged and weighted by class frequency metrics. These results show that tf-idf values are good features when classifying huge documents.

Table 2.5 compares models trained with and without pages labelled as *Others*, thought to be less informative by the Court experts. The classes' F_1 scores show great variability, with numbers ranging from 0 to 100 in both cases. That is not surprising, considering the number of examples for the themes with extreme scores, which is between 0 and 4. Due to the small number of samples, such scores are not very reliable.

That being said, the overall results oppose the domain expert intuition, since the weighted and average F_1 scores for the model trained with *Others* pages were 6.77 p.p. and 12.42 p.p. higher, respectively, than the model trained without such pages. That is, contrary to domain knowledge expectations, the data is useful for the task and should not be disregarded.

Table 2.4: F_1 score (in %) of our methods for theme classification on the test sets. A baseline that always assigns all themes yields an F_1 score weighted by class frequencies of 41.17 /40.87/10.87 and an average F_1 score of 5.48/5.49/6.52 on BVic, MVic, SVic, respectively.

	BVic			MVic			SVic		
Themes	NB	SVM	XGBoost	NB	SVM	XGBoost	NB	SVM	XGBoost
0	81.63	87.35	**90.70**	79.50	88.85	**92.41**	49.90	**72.29**	69.71
5	17.95	92.47	**94.15**	18.73	79.05	**85.50**	30.22	**84.79**	82.87
6	65.85	61.65	**77.84**	37.45	36.52	**76.81**	21.93	63.11	**77.03**
26	60.38	92.06	**93.33**	14.59	36.48	**94.74**	12.75	**97.44**	94.44
33	30.03	46.32	**77.17**	8.35	14.42	**78.62**	30.71	57.78	**74.65**
139	61.82	81.25	**90.57**	17.54	74.67	**92.59**	14.95	88.89	**94.34**
163	77.38	75.41	**86.09**	25.05	76.19	**88.00**	73.86	86.08	**94.67**
232	40.93	44.64	**69.33**	27.63	13.90	**55.12**	37.32	65.00	**65.08**
313	47.42	58.56	**72.55**	31.11	43.37	**80.77**	60.22	76.12	**82.69**
339	23.17	52.12	**74.47**	20.62	45.84	**77.04**	26.73	74.38	**86.06**
350	73.27	55.26	**86.96**	73.27	12.05	**89.58**	85.06	52.94	**90.11**
406	57.41	44.44	**85.71**	20.27	10.41	**85.71**	55.81	46.15	**84.93**
409	74.42	79.12	**86.25**	29.03	72.64	**90.68**	91.14	90.91	**95.48**
555	39.02	65.06	**83.33**	0.00	17.06	**84.75**	47.06	52.46	**88.89**
589	77.97	82.01	**88.00**	35.02	63.44	**88.71**	82.05	90.16	**90.76**
597	**96.77**	90.91	96.55	53.57	90.91	**96.55**	85.71	88.24	**96.77**
634	89.87	90.91	**95.48**	70.24	89.29	**94.19**	92.81	93.08	**95.42**
660	51.23	74.14	**89.00**	35.30	80.39	**90.07**	36.41	91.10	**93.51**
695	93.27	**97.65**	96.65	95.37	**98.13**	96.68	96.52	**98.49**	96.94
729	**100.00**	**100.00**	97.78	62.07	**95.65**	93.02	63.16	**100.00**	93.33
766	21.88	73.21	**77.65**	21.82	76.64	**82.61**	19.81	81.08	**86.67**
773	68.03	96.40	**97.06**	61.54	95.71	**98.55**	81.30	**94.03**	93.13
793	66.67	84.52	**92.96**	28.26	86.23	**91.43**	26.59	87.80	**90.79**
800	87.70	98.42	**98.73**	87.34	98.41	**98.62**	69.86	**92.71**	91.10
810	62.28	88.72	**95.32**	23.89	92.16	**94.87**	21.06	**95.62**	94.69
852	64.67	82.61	**87.34**	54.40	76.68	**89.74**	49.08	89.41	**92.31**
895	25.10	63.68	**89.66**	14.64	94.08	**98.32**	24.07	92.17	**95.93**
951	94.74	**100.00**	99.54	39.04	98.21	**98.62**	57.36	**99.50**	95.29
975	86.15	91.67	**94.44**	15.62	68.69	**91.43**	41.61	**89.74**	**89.74**
Weighted	69.55	82.35	**89.57**	60.62	81.37	**90.72**	48.75	82.31	**86.34**
Average	63.35	77.61	**88.43**	37.97	66.42	**88.82**	51.21	82.46	**88.87**

Table 2.5: F_1 score (in %) of a XGBoost trained without and with *Others* pages on BVic test set filtered to include only lawsuits with at least one page not classified as *Others*.

Themes	Without	With	Count
0	91.15	**92.55**	832
5	**93.33**	85.71	8
6	70.00	**81.82**	13
33	**0.00**	**0.00**	3
139	**50.00**	0.00	2
163	90.65	**91.43**	67
232	69.77	**80.00**	23
313	**77.78**	70.00	11
339	49.32	**70.89**	48
350	**100.00**	**100.00**	1
406	**0.00**	**0.00**	4
409	87.58	**89.93**	71
555	54.55	**83.33**	7
589	86.96	**92.63**	47
597	90.91	**90.91**	6
634	**95.83**	90.57	25
660	33.80	**86.05**	49
695	89.29	**92.86**	29
729	**100.00**	96.97	17
766	57.14	**66.67**	10
773	**94.55**	**94.55**	29
793	**0.00**	**0.00**	4
800	80.40	**97.78**	115
810	76.19	**87.50**	44
852	82.05	**92.68**	19
895	0.00	**100.00**	2
Weighted	84.55	**90.27**	1,486
Average	66.20	**74.42**	

2.1.6 Summary

This section introduces the VICTOR Dataset, a corpus of legal documents from Brazil's Supreme Court. VICTOR features two types of tasks: document type classification, with six disjoint document categories; and theme assignment, a multi-label problem with 29 different classes. The data is available in three versions: BVic, containing data for the theme assignment task; MVic, containing only type-labelled documents, for both tasks; and SVic, a subsample of MVic.

We also establish benchmarks for the presented tasks, comparing textual and sequential data representations. Our experiments with CRF post-processing show that the sequential nature of the suits may be leveraged to improve document type classification. Furthermore,we find that tf-idf features are good descriptors of long texts, where common deep learning approaches are not easily applicable.

In the next section, we will present a corpus of official texts with document source annotation and examine another text classification task. While here we only make use of labelled documents, there we will leverage unlabelled examples to create a more robust model.

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

Official Gazettes are a rich source of relevant information to the public. Their careful examination may lead to the detection of frauds and irregularities that may prevent mismanagement of public funds. This section presents a dataset composed of documents from the Official Gazette of Brazil's Federal District, containing both samples with document source annotation and unlabelled ones. We train, evaluate and compare a transfer learning based model that uses Universal Language Model Fine-tuning (ULMFiT) [3] to traditional Bag-Of-Words (BOW) models that use SVM and Naïve Bayes as classifiers. We find the SVM to be competitive, its performance being marginally worse than the ULMFiT while having much faster train and inference time and being less computationally expensive. Finally, we conduct ablation analysis to assess the performance impact of the ULMFiT parts.[9]

[9] An early version of this section has been published in: Luz de Araujo, P. H. et al. Inferring the source of official texts: can SVM beat ULMFiT? [56].

2.2.1 Introduction

Government Gazettes are a great source of information of public interest. These government maintained periodical publications disclose a myriad of matters, such as contracts, public notices, financial statements of public companies, public servant nominations, public tenderings, public procurements and others. Some of the publications deal with public expenditures and may be subject to frauds and other irregularities.

That said, it is not easy to extract information from Official Gazettes. The data is not structured, but available as natural language texts. In addition, the language used is typically from the public administration domain, which can further complicate information extraction and retrieval by general-domain applications.

As we previously stated, Natural Language Processing (NLP) and Machine Learning (ML) techniques are great tools for obtaining information from official texts. NLP has been used to automatically extract and classify relevant entities in court documents [32, 33]. Other works [34, 35, 36, 37] explore the use of automatic summarization to mitigate the amount of information legal professional have to process. Text classification has been utilized for decision prediction [42, 41], area of legal practice attribution [43] and fine-grained legal-issue classification. Some effort has been applied to the processing of Brazilian legal documents [47, 57, 17], as we previously discussed (Section 2.1).

In this section, we aim to identify which public entity originated documents fom the Official Gazette of the Federal District. This is a first step in the direction of structuring the information present in Official Gazettes in order to enable more advanced applications such as fraud detection. Even though it is possible to extract the public entity that produced the document by using rules and regular expressions, such approach is not very robust: changes in document and phrase structure and spelling mistakes can greatly reduce its effectiveness. A machine learning approach may be more robust to such data variation.

Due to the small number of samples in our dataset, we explore the use of transfer learning for NLP. We choose ULMFiT [3] as the method due to it being less resource-intensive than other state-of-the-art approaches such as Bidirectional Encoder Representations from Transformers (BERT) [2] and Generative Pre-trained Transformer 2 (GPT-2) [14]. Our main contributions[10] are:

1. Making available to the community a dataset with labelled and unlabelled Official Gazette documents.

[10] Resources (data, code and trained models) from this section are available at `https://cic.unb.br/~teodecampos/KnEDLe/propor2020/`

2. Training, evaluating and comparing a ULMFiT model to traditional bag-of-word models.

3. Performing an ablation analysis to examine the impact of the ULMFiT steps when trained on our data.

2.2.2 The DODF dataset

The DODF [11] data consists of 2,652 texts extracted from the Official Gazette of the Federal District[12]. Handcrafted regex rules were used to extract some information from each sample, such as publication date, section number, public body that issued the document and title. 797 of the documents were manually examined, from which 724 were found to be free of labelling mistakes. These documents were produced by 25 different public entities. We filter the samples with entities with less than three samples, since this would mean no representation for the public body in either the training, validation or test set. As a result, we end up with 717 labelled examples from 19 public entities.

We then split these samples and the 1,928 unverified or incorrectly labelled texts into two separate datasets. The first for classification of public entity that produced the document and the other for the unsupervised training of a language model.

The classification dataset is formed by 717 pairs of document and its respective public entity of origin. We randomly sample 8/15 of the texts for the training set, 2/15 for the validation set and the remainder for the test set, which results in 384, 96 and 237 documents in each set, respectively. Figure 2.8 shows the class distribution in each set. The data is imbalanced: *Segurança Pública*, the most frequent class, contains more than 140 samples, while the least frequent classes are represented by less than 5 documents. We handle this by using F_1 score as the metric for evaluation and trying model-specific strategies to handle imbalance, as we discuss in Section 2.2.4.

Two of the 1,928 texts in the language model dataset were found to be empty and were dropped. From the remaining 1,926, 20% were randomly chosen for the validation set. The texts contain 984,580 tokens in total; after the split, there are 784,260 in the training set and 200,320 in the validation set. In this case we choose to not build a test set since we are not interested in an unbiased evaluation of the language model performance. The data is automatically labelled as a standard language model task where the label of each token is the following token in the sentence.

[11] *Diário Oficial do DF*—Official Gazette of the Federal District.

[12] Published at `https://www.dodf.df.gov.br/`.

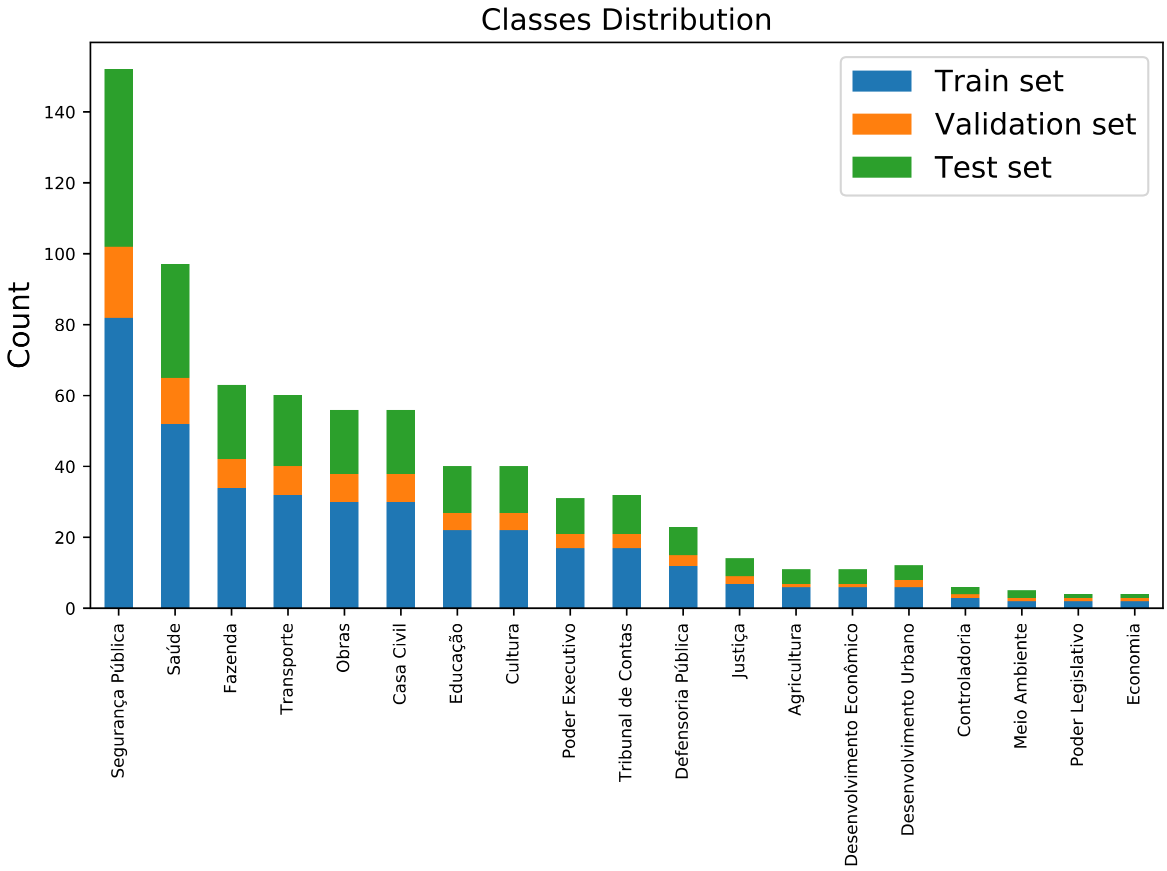


Figure 2.8: Class counts for each DODF split.

2.2.3 The models

Here we describe the transfer learning based approach to text classification used to classify the documents, the BOW method used as a baseline and the preprocessing employed for both approaches.

Preprocessing

We first lowercase the text and use SentencePiece [58] to tokenize it. We chose SentencePiece because that was the tokenizer used for the pre-trained language model (more about that on Section 2.2.3), so using the same tokenization was fundamental to preserve vocabulary. We use the same tokenization for the baseline methods to establish a fair comparison of the approaches.

In addition, we add special tokens to the vocabulary to indicate unknown words, padding, beginning of text, first letter capitalization, all letters capitalization, character repetition and word repetition. Even though the text has been lowercased, these tokens preserve the capitalization information present in the original data. The final vocabulary is composed of 8,552 tokens, including words, subwords, special tokens and punctuation.

Baseline

For the baseline models, we experiment with two different BOW text representation methods: tf-idf values and token counts. Both methods represent each document as a v-dimensional vector, where v is the vocabulary size. In the first case, the i-th entry of the vector is the tf-idf value of the i-th token in the vocabulary, while in the second case that value is simply the number of times the token appears in the document. Tf-idf values are computed according to the following equations:

$$\text{tf-idf}(t,d) = \text{tf}(t,d) \times \text{idf}(t) \tag{2.2}$$

$$\text{idf}(t) = \log \frac{1+n}{1+\text{count}(t)} + 1\,, \tag{2.3}$$

where $\text{tf}(t,d)$ is the frequency of term t in document d, n is the total of documents in the corpus, and $\text{count}(t)$ is the number of documents that contain term t. All document vectors are normalised to have unit Euclidean norm.

We use the obtained BOW to train a shallow classifier. We experiment with both SVM [59] with linear kernel and NB classifiers.

Transfer Learning

We use ULMFiT [3] to leverage information contained in the unlabelled language model dataset. This method of inductive transfer learning was shown to require much fewer labelled examples to match the performance of training from scratch.

ULMFiT comprises three stages:

Language model pre-training We use a bidirectional Portuguese language model[13] previously trained on a Wikipedia corpus composed of 166,580 articles, with a total of 100,255,322 tokens. The tokenization used was the same as ours. The model architecture consists of a 400-dimensional embedding layer, followed by four Quasi-Recurrent Neural Network (QRNN) [60] layers with 1550 hidden parameters each and a final linear classifier on top. QRNN layers alternate parallel convolutional layers and a recurrent pooling function, outperforming LSTMs of same hidden size while being faster at trainining time and inference.

Language model fine-tuning We fine-tune the forward and backward pre-trained general-domain Portuguese language models on our unlabelled dataset, since the latter comes from the same distribution and the classification task data, while the former does not. As in the ULMFiT paper, we use discriminative fine-tuning [3], where instead of using the same learning rate for all layers of the model, different learning rates are used for different layers. We employ cyclical learning rates [61] with cosine annealing to speed up training.

Classifier fine-tuning To train the document classifier, we add two linear blocks to the language models, each block composed of batch normalization [62], dropout [63] and a fully-connected layer. The first fully-connected layer has 50 units and ReLU [64] activation, while the second one has 19 units and is followed by a softmax activation that produces the probability distribution over the classes. The final prediction is the average of the forward and backwards models. The input to the linear blocks is the concatenation of the hidden state of the last time step $\mathbf{h}_T$ with the max-pooled and the average-pooled hidden states of as many time steps as can be fit in GPU memory $\mathbf{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_T\}$. That is, the input to the linear blocks $\mathbf{h}_c$ is:

$$\mathbf{h}_c = \text{concat}(\mathbf{h}_t, \text{maxpool}(\mathbf{H}), \text{averagepool}(\mathbf{H}))\,. \tag{2.4}$$

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

2.2.4 Experiments

Here we describe the training procedure and hyperparameters used. All experiments were executed on a Google Cloud Platform n1-highmem-4 virtual machine with a Nvidia Tesla P4 GPU, which has 8 GB of internal memory.

Baseline

To find the best set of hyperparameter values we use random search and evaluate the model on the validation set. Since we experiment with two classifiers (SVM and NB) and two text vectorizers (tf-idf values and token counts), we have four model combinations: tf-idf and NB, tf-idf and SVM, token counts and NB; and token counts and SVM. For each of these 4 scenarios we train 100 models, each iteration with random hyperparameter values, as detailed below.

Vectorizers For both the tf-idf and token counts vectorizers we tune the same set of hyperparameters: n-gram range (only unigrams, unigrams and bigrams, unigrams to trigrams), maximum document frequency token cutoff (50%, 80% and 100%), minimum number of documents for token cutoff (1, 2 and 3 documents).

NB We tune the smoothing prior α on a exponential scale from 10^{-4} to 1. We also choose between fitting the prior probabilities, which could help with the class imbalance, and just using a uniform prior distribution.

SVM In the SVM case, we tune two hyperparameters. We sample the regularization parameter C from an exponential scale from 10^{-3} to 10. In addition, we choose between applying weights inversely proportional to class frequencies to compensate class imbalance and giving all classes the same weight.

Transfer Learning

To tune the best learning rate in both the language model fine-tuning and classifier training scenarios, we use the learning rate range test [65], where we run the model through batches while increasing the learning rate value, choosing the learning rate value that corresponds to the steepest decrease in validation loss. We use Adam [66] as the optmiser.

We fine-tune the top layer of the forward and backwards language models for one cycle of 2 epochs and then train all layers for one cycle of 10 epochs. We use a batch size of 32 documents, weight decay [67] of 0.1, backpropagation through time of length 70

Table 2.6: Classification results (in %) on the test set.

Class	NB	SVM	F-ULMFiT	B-ULMFiT	F+B-ULMFiT	Count
Casa Civil	66.67	78.95	80.00	82.35	**88.24**	18
Controladoria	80.00	80.00	**100.00**	**100.00**	**100.00**	2
Defensoria Pública	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	8
Poder Executivo	80.00	85.71	78.26	**90.91**	86.96	10
Poder Legislativo	66.67	**100.00**	66.67	66.67	**100.00**	1
Agricultura	50.00	**66.67**	57.14	50.00	57.14	4
Cultura	**91.67**	**91.67**	**91.67**	**91.67**	**91.67**	13
Desenv. Econômico	**66.67**	**66.67**	**66.67**	**66.67**	**66.67**	4
Desenv. Urbano	75.00	75.00	75.00	**85.71**	75.00	4
Economia	66.67	**100.00**	**100.00**	**100.00**	**100.00**	1
Educação	76.19	**91.67**	81.48	75.00	88.00	13
Fazenda	90.48	90.48	95.00	95.24	**97.56**	21
Justiça	**75.00**	66.67	60.00	66.67	66.67	5
Obras	88.24	**90.91**	88.24	76.92	85.71	18
Saúde	92.75	92.31	92.31	94.12	**95.52**	32
Segurança Pública	**98.99**	94.34	94.34	97.09	94.34	50
Transporte	94.74	**97.56**	92.31	92.31	**97.56**	20
Meio Ambiente	**100.00**	**100.00**	66.67	0.00	0.00	2
Tribunal de Contas	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	11
Average F_1	82.09	**87.82**	83.46	80.6	83.74	237
Weighted F_1	88.68	90.49	88.90	88.88	**90.88**	237
Accuracy	88.61	90.72	89.03	89.45	**91.56**	237

and dropout probabilities of 0.1, 0.6, 0.5 and 0.2 applied to embeddings inputs, embedding outputs, QRNN hidden-to-hidden weight matrix and QRNN output, respectively, following previous work [3].

In the case of the backward and forward classifiers, in order to prevent catastrophic forgetting by fine-tuning all layers at once, we gradually unfreeze [3] the layers starting from the last layer. Each time we unfreeze a layer we fine-tune for one cycle of 10 epochs. We use a batch size of 8 documents, weight decay of 0.3, backpropagation through time of length 70 and the same dropout probabilities used for the language model fine-tuning scaled by a factor of 0.5.

Similarly to the SVM experiments, in order to handle data imbalance we try applying weights inversely proportional to class frequencies. Nevertheless, this did not contribute to significant changes in classification metrics.

2.2.5 Results

Table 2.6 reports, for each model trained, test set F_1 scores for each class. Due to the small size of the classification dataset, some class-specific scores are noisy because of their rarity, so we also present the average and weighted by class frequency F_1 values and the model accuracy. For the baseline models, we present results using the tf-idf vectorizer (unigrams to trigrams, 50% maximum frequency cutoff, minimum cutoff of at least 1 document, which generated a vocabulary of 144,857 tokens) with the NB classifier and the count vectorizer (unigrams to trigrams, 50% maximum frequency cutoff, minimum cutoff of at least 3 documents, which generated a vocabulary of 29,646 tokens) with the SVM classifier. These combinations were the best performing on the validation set. F-ULMFiT, B-ULMFiT and F+B-ULMFiT indicate the forward ULMFiT model, the backward counterpart and their ensemble, respectively.

All models performed better than a classifier that simply chooses the most common class, which would yield average and weighted F_1 scores of 7.35% and 1.83% and an accuracy of 21.10%. The SVM and ULMFiT models outperformed the NB classifier across almost all categories. All models seem to achieve good results, with weighted F_1 scores and accuracies approaching 90.00%, though we do not have a human performance benchmark for comparison.

Despite the SVM average F_1 score being higher than the ULMFiT's, the latter has greater weighted F_1 score and accuracy, with a corresponding reduction of 9.05% on test error rate. That being said, the SVM has some advantages. First, it is much faster to train. While the SVM took less than two seconds to train, the ULMFiT model took more than half an hour—not counting the language model pre-training, which took hours[14]. In addition, the ULMFiT approach greatly depends on GPU availability, otherwise training would take much longer.

Furthermore, SVM training is very straightforward, while the transfer learning scenario requires three different steps with many parts that need tweaking (gradual unfreezing, learning rate schedule, discriminative fine-tuning). Consequently, not only the ULMFiT model has more hyperparameters to be tuned, each parameter search iteration is computationally expensive—the time it takes to train one ULMFiT model is enough to train more than 1,000 SVM models with different configurations of hyperparameters.

Ablation analysis

Here we analyse the individual impact of ULMFiT's parts on our data. We do so by running experiments on four different scenarios. We use the same hyperparameters as in the complete ULMFiT case and train for the same number of iterations in order to establish a fair comparison. Table 2.7 presents the results and the difference between the scenario result and the original performance, taking into consideration if it is the forward, backward or ensemble case.

[14] `https://github.com/piegu/language-models/blob/master/lm3-portuguese.ipynb`

Table 2.7: Ablation scenarios results (in %) on the test set. Metrics are compared to the corresponding full ULMFiT model (forward, backward or ensemble).

Model	Average F_1	Weighted F_1	Accuracy
No gradual unfreeezing (f)	86.57 (+3.11)	88.80 (-0.10)	89.03 (+0.00)
No gradual unfreeezing (b)	88.05 (+7.45)	92.30 (+3.42)	92.41 (+2.96)
No gradual unfreeezing (f+b)	89.18 (+5.44)	92.57 (+1.69)	92.83 (+1.27)
Last layer fine-tuning (f)	65.59 (-17.87)	76.51 (-12.39)	77.64 (-11.39)
Last layer fine-tuning (b)	60.93 (-19.67)	76.22 (-12.66)	78.06 (-11.39)
Last layer fine-tuning (f+b)	68.01 (-15.73)	77.92 (-12.96)	79.32 (-12.24)
No LM fine-tuning (f)	39.61 (-43.85)	58.79 (-30.11)	63.71 (-25.32)
No LM fine-tuning (b)	39.81 (-40.79)	61.80 (-27.08)	65.82 (-23.63)
No LM fine-tuning (f+b)	44.32 (-39.42)	66.33 (-24.55)	69.26 (-22.30)
Direct transfer (f)	11.46 (-72.00)	24.59 (-64.31)	34.18 (-54.85)
Direct transfer (b)	12.29 (-68.31)	27.35 (-61.53)	38.40 (-51.05)
Direct transfer (f+b)	12.36 (-71.38)	26.35 (-64.53)	37.97 (-53.59)

No gradual unfreezing This scenario's training procedure is almost identical to the previously presented, with the exception that gradual unfreezing is not used. In the classifier fine-tuning step though, we instead fine-tune all layers at the same time. This was the least contributing to the performance—in fact, the model trained without gradual unfreezing performed better than the standard model across all metrics. This is surprising, since gradual freezing was shown to be beneficial in the paper that proposed ULMFiT [3]. As such, this finding may be an artifact of the small size of our test data.

Last layer fine-tuning This scenario is similar to the previous one in the sense that we do not perform gradual unfreezing. But while there we fine-tuned all layers, here we treat the network as a feature extractor and fine-tune only the classifier. We see a sharp decrease in performance across all metrics, suggesting that the QRNN network, even though the language model was fine-tuned on domain data, does not perform well as a feature extractor for document classification. That is, to train a good model it is imperative to fine-tune all layers.

No language model fine-tuning Here we skip the language model fine-tuning step and instead train the classifier directly from the pre-trained language model, using gradual unfreezing just like in the original model. This results in a great decline in performance, with decreases ranging from about 20 to more than 40 percentual points. Therefore, for our data, training a language model on general domain data is not enough; language model fine-tuning on domain data is essential. This may be due to differences in word distribution between general and official text domains.

Direct transfer In this scenario we go one step further than in the previous one: we start from the pre-trained language model and do not fine-tune it. They differ because in the classifier fine-tuning step we do not perform gradual unfreezing, but train all layers at the same time. This results in a even greater performance decrease. The lack of gradual unfreezing here is much more dramatic than in the first scenario. We hypothesize that the language model fine-tuning may mitigate the effects or decrease the possibility of catastrophic forgetting.

Averaging forward and backward predictions In almost all cases, averaging the forward and backward models predictions results in more accurate results than either of the single models. One possible way of further experimenting is trying other methods of combining the directional outputs.

2.2.6 Summary

This section examines the use of ULMFiT, an inductive transfer learning method for natural language applications, to identify the public entity that originated Official Gazette texts. We compare the performance of ULMFiT with simple BOW baselines and perform an ablation analysis to identify the impact of gradual unfreezing, language model fine-tuning and the use of the fine-tuned language model as a text feature extractor.

Despite being a state-of-the-art technique, the use of ULMFiT corresponds to a small increase in classification accuracy when compared to the SVM model. Considering the faster training time, simpler training procedure and easier parameter tuning of SVM, this traditional text classification method is still competitive with modern deep learning models. A potential reason for that is that word order is not so important for the presented task.

Finally, our ablation analysis shows that language model fine-tuning is essential to the transfer learning approach. That said, it also suggests that language models, even after fine-tuned on domain data, are not good feature extractors and should be trained also on classification data.

2.3 Conclusions

In this chapter we have proposed two text classification datasets with different domains and particularities. We have analysed both single-label and multi-label classification of texts ranging from small documents to large lawsuits, using both deep neural network architectures and BOW models. We have found that in the tasks presented, where word order is not of utmost importance, the SVM is still competitive when compared to state-of-the-art models—and other shallow classifiers, such as XGBoost [55], could perform even better. In the next chapter we will focus on another technique used to extract document-level knowledge: topic modelling.

Chapter 3

Topic Modelling

Topic modelling, similarly to text classification, is concerned with extracting knowledge at the document-level. That said, while the latter aims to assign samples to predefined categories in a supervised way, the former discovers abstract topics present in the corpus in a unsupervised manner. The topics are mere probability distribution over words, so that human understanding is needed to interpret and label the topics. In addition, each document is modeled as a distribution over topics. This is often used as a measure of similarity between documents, which can help organising massive collections of documents and understanding their major themes.

Such distributions over topics can also be viewed as vectors of text features and used for text classification tasks instead of traditional BOW models. In this chapter, we shed new light to one of the datasets previously presented to examine both uses of topic modelling: as a tool for finding subject matters present in the corpus and as a way to create text representations for downstream tasks.

In Section 3.1 we propose the use of Latent Dirichlet Allocation to model Extraordinary Appeals from Brazil's Supreme Court. We first examine the "orthodox" use of topic models by analysing the topics obtained and labelling them. Then we turn to the less common usage, where we use the topic distributions constructed in the previous step to train a general repercussion theme classifier.

3.1 Topic modelling Brazilian Supreme Court lawsuits

The present work proposes the use of Latent Dirichlet Allocation (LDA) to model Extraordinary Appeals received by Brazil's Supreme Court. The data consist of the corpus described in 2.1.3, containing 45,532 lawsuits manually annotated by the Court's experts with theme labels, a multi-class and multi-label classification task. We initially train models with 10 and 30 topics and analyse their semantics by examining each topic's most relevant words and their most representative texts, aiming to evaluate model interpretability and quality. We also train models with 30, 100, 300 and 1,000 topics, and use them to generate a feature vector for each appeal and train a lawsuit theme classifier. We compare traditional Bag-Of-Words (BOW) approaches (word counts and tf-idf values) with the topic-based text representation to assess if the latter is viable as a text representation method for classification purposes. Our topics semantic analysis demonstrate that our models with 10 and 30 topics were capable of capturing some of the legal matters discussed by the Court. In addition, our experiments show that the model with 300 topics was the best text vectorizer and that the interpretable, low dimensional representations it generates achieve good classification results.

3.1.1 Introduction

As we describe in 2.1.1, Brazil's court system suffers from an excessive amount of lawsuits [23]. Natural Language Processing (NLP) and Machine Learning (ML) techniques can contribute to a quicker, cheaper and more efficient analysis of legal proceedings and as a result help promote greater effectiveness and democratization of justice. Some works already explore the use of artificial intelligence in the context of Brazil's courts [47, 17, 57]. That being said, we are not aware of publications regarding the topic modeling of Brazilian lawsuits.

Topic Models are a family of statistical models used to discover in an automatic and unsupervised manner themes (topics) present in a collection of documents [68]. The topics are obtained from the statistical analysis of the words that comprise the documents. Since annotations and labelling of documents are not needed, Topic Models enable the organisation, exploration and indexing of massive amounts of data in a scale that could be prohibitively expensive if human made. The trained models may also be used for downstream tasks such as sentiment analysis [69] and document classification [70]. In addition, the approach is not restricted to text data and may be used to model genomic data, images and social networks [68].

In this section, we employ Latent Dirichlet Allocation (LDA) to model Extraordinary Appeals (*Recursos Extraordinários*—RE) received by Brazil's Supreme Court (*Supremo Tribunal Federal*—STF). Each suit has been manually annotated by the Court's employees to include information on its General Repercussion (*Repercussão Geral*) themes. This is a multi-label classification task, which we will further discuss in 3.1.3. Our contributions are:

1. The analysis of the semantics of each topic from models with 10 and 30 topics trained on the STF data.

2. The evaluation of the use of topic distribution vectors as a form of suit representation for theme classification, comparing it with the performance of traditional text representation Bag-Of-Words approaches that use word counts or tf-idf values. We experiment with models of 10, 30, 100, 300 and 1,000 topics. Our aim here is not to beat the SOTA of for the data, but to assess the viability of topic models as a text representation method for classification.

The rest of the section is organised as follows. First, we briefly review Topic Model literature and NLP applied to the legal domain approaches (3.1.2). Then we describe the dataset (3.1.3) and the model employed (3.1.4). Following that, we report our experiments (3.1.5) and present and discuss the results (3.1.6). Finally, we present our final considerations (3.1.7).

3.1.2 Related work

Topic Models

Topic Models have been an area of research since 1990, when Deerwester et al. [71] proposed Latent Semantic Indexing (LSI). The method uses Singular Value Decomposition (SVD) to factorize a matrix of term-document co-occurrence values to construct a "semantic" space where terms and documents closely associated are near one another. The method is further explored by Hofmann [72], who introduced probabilistic LSI. Like LSI, PLSI decomposes a co-occurrence matrix, but while the former uses a linear algebra approach, the latter method is statistical, modeling the document-word co-occurrence probability as a mixture of conditionally independent multinomial distributions. On the other hand, PLSI has some weaknesses, such as the linear growth of the parameters with the size of the corpus, which causes overfitting issues, and the lack of procedure to assign probability to a document not seen in the training set.

To overcome PLSI weaknesses, Blei et al. [73] proposed Latent Dirichlet Allocation. The authors show that LDA can be used for a range of tasks, such as document modeling, text classification and collaborative filtering, outperforming approaches based on unigrams and PLSI.

Since then, the study of extensions of LDA by relaxing some of its assumptions has been an active area of research [68]. For example, by relaxing the assumption that the order of the documents can be neglected, Blei and Lafferty [74] propose Dynamic Topic Models, capable of modeling the time evolution of topics in a corpus.

Natural Language Processing and Topic Models in Legal Text

Efforts have been made to apply NLP and ML techniques to legal text. Natural Language Processing has been used to automatically extract and classify relevant entities in court documents [32, 33, 17]. Other works [34, 35, 36, 37] focus on using automatic summarization to reduce the amount of information legal professionals have to process. Document classification has been explored for decision prediction [42, 41], area of legal practice attribution [43] and fine-grained legal-issue classification [44].

Regarding LDA, the method has been employed to model legal corpora. Carter et al. [38] model documents from the Australian High Court; Remmits [39] models decisions from the Supreme Court of the Netherlands; O'Neill et al. [40] used LDA to explore British legislative texts.

Some works explore the processing of Brazilian legal documents. Correia da Silva et al. [47] use a CNN to classify STF's documents. De Vargas Feijó and Moreira [57] introduce a dataset for decision summarization. Luz de Araujo et al. [17] built a manually annotated corpus for named entity recognition and classification with legislation and legal decision classes. On the other hand, we are not aware of publications studying topic modeling of Brazilian legal corpora.

3.1.3 The dataset

We use the same dataset described in 2.1.2 [15], which contains 45,532 Extraordinary Appeals. Each instance is a legal proceeding as it is received by the STF, that is, before it is processed and judged. In addition, a lawsuit is represented as an ordered sequence of pages containing text.

The dataset contains manual annotation that assigns to each lawsuit one or more general repercussion[1] themes. More specifically, the options are the 28 most important themes according to the STF, each one identified by a unique integer[2]; e.g., theme 6 deals with the State's duty to supply costly medications to citizens who suffer from serious diseases and are not able to buy them. The integer 0 identifies the instances that contain at least one theme that does not belong to any of those 28 classes. It follows that theme assignment is a multi-label classification task.

The data is divided into train/validation/test splits containing 70%/15%/15% of all suits, respectively. The theme distribution is the same in all splits as figure 3.1 shows.

[1] An appeal must have general repercussion to be considered by the STF. This means that lawsuit must relate to relevant economic, political, social or legal issues that exceed the interests of the parties.

[2] A list of all themes is available at `http://www.stf.jus.br/portal/jurisprudenciaRepercussao/abrirTemasComRG.asp`.

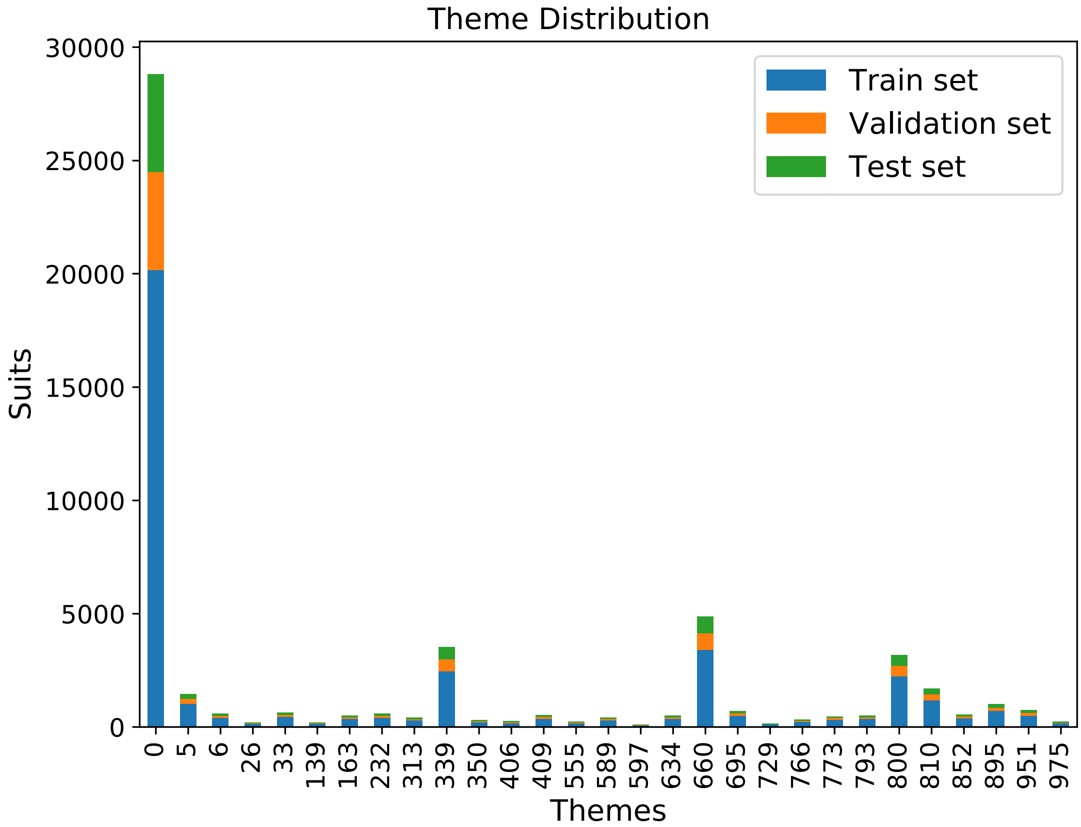


Figure 3.1: Theme counts.

The following preprocessing steps were applied to the raw text: lower-casing, removal of stop words and alphanumeric tokens, email and URL tokenization, and identification of simple law citations; e.g., we change *Lei* (law) 11.419 to LEI_11419.

3.1.4 The model

Inspired by previous attempts to model different kinds of legal text [38, 39, 40], we choose Latent Dirichlet Allocation [73] as the method for topic generation. We use the following terminology [73]:

- A *word* is the discrete unit of data defined as an entry of a vocabulary indexed by $\{1, \ldots, \mathcal{V}\}$. Each word is represented as one-hot encoded vector; i.e., when using superscript to denote vector components, the v-th word of the vocabulary is represented by a $\mathcal{V}$-dimensional vector $\mathbf{w}$ such that $\mathbf{w}^v = 1$ and $\mathbf{w}^u = 0$ for $u \neq v$.
- A *document* is a sequence of n words denoted by $W = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$.
- A *corpus* is a set of m documents denoted by $D = \{\mathbf{W}_1, \ldots, \mathbf{W}_m\}$.

LDA is a probabilistic generative model of a corpus, where each document is represented as a random mixture over latent topics. Each topic is in turn a distribution over words. That is, LDA assumes the following generative process for a corpus D of m documents of length n_i, $i \in [1, \ldots, m]$, assuming a fixed set of k topics:

1. $\boldsymbol{\theta}_i$, $i \in \{1, \ldots, m\}$, the topic distribution of document i, is chosen from a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha})$
2. $\boldsymbol{\phi}_j$, $j \in \{1, \ldots, k\}$, the word distribution of topic j, is chosen from a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\beta})$.
3. For each word position (i, j), $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n_j\}$:
   (a) A topic $\mathbf{z}_{i,j} \sim \mathrm{Multinomial}(\boldsymbol{\theta}_i)$ is chosen.
   (b) A word $\mathbf{w}_{i,j} \sim \mathrm{Multinomial}(\boldsymbol{\phi}_{\mathbf{z}_{i,j}})$ is chosen.

Given this generative assumption, the LDA procedure assigns: a topic distribution for each document, a topic for each word in each document and a word distribution for each topic.

3.1.5 Experiments

Model training for exploratory analysis

We use LDA to perform an exploratory analysis of the data aiming to understand its most relevant topics. We train two models on the training split of the data, one with 10 topics and the other with 30. Since the whole data does not fit into memory, we use the algorithm proposed by [75] for the online training of LDA models, based on stochastic optmisation with gradient steps.

To select the most informative words, we restrict our vocabulary to the words that appear in at least 50 lawsuits of the training set and in no more than 50% of them. In addition, we filter words with only one letter, with the intuition that they probably do not help with topic interpretability. The obtained vocabulary contains 81,418 entries.

We use mini-batches of 4,096 suits, with a maximum number of 400 iterations per mini-batch, and train for 4 epochs. The hyperparameters were chosen empirically and were sufficient for the convergence of most lawsuits in the training set.

Topic distribution as text representation

In order to examine the use of topics for text representation, we use LDA as a lawsuit feature extractor; that is, the topic distribution of each lawsuit is used as its vector representation and fed to a classifier to predict general repercussion themes. We run experiments with models of 10, 30, 100, 300 and 1,000 topics, using eXtreme Gradient Boosting (XGBoost) [55]—as the classifier.

We compare the topic representation with two traditional bag-of-words representations: i) tf-idf values and ii) word counts. To establish a fair comparison, all models use the same vocabulary. Since we have a multi-label task, we employ a One-vs-All approach where we train a binary classifier for each theme and the final classification is the aggregation of all predictions. Formally, let C be the set of all themes, t a threshold value, $f_c(\cdot)$ the decision function of the classifier for class c, and l a lawsuit:

$$\forall c \in C, \text{assign } c \text{ to } l \text{ if } f_c(l) \geq t\,. \tag{3.1}$$

We set 0.5 as the threshold value.

Finally, we use the validation set to tune the following XGBoost hyperparameters through random search: number of trees, maximum depth and shrinkage factor.

All results are reported on the test set unless otherwise stated. As a baseline method we choose a classifier that assigns all themes to any input, which achieves an F_1 score weighted by class frequency of 41.17 and an average F_1 score of 5.48.

3.1.6 Results

Topic analysis

In order to evaluate the topic quality of the models with 10 and 30 topics we examine the most relevant words and lawsuits from each topic and assign it a label [76]. Table 3.1 presents the results of the labelling process. For each topic we show its four most relevant words, where relevance is defined [77] as

$$r(\mathbf{w}, \mathbf{z}|\lambda) = \lambda \log P(\mathbf{w}|\mathbf{z}) + (1 - \lambda) \log \frac{P(\mathbf{w}|\mathbf{z})}{P(\mathbf{w})}\,, \tag{3.2}$$

and the parameter λ ($0 \leq \lambda \leq 1$) determines weight given to the probability of term $\mathbf{w}$ given topic $\mathbf{z}$ relative to the ratio between that probability and the marginal probability of $\mathbf{w}$ on the whole corpus. For each topic, through manual inspection, we select the value with the most descriptive top words, which have been translated to English, except in the case of acronyms and names, which are shown in italic.

Table 3.1: Topic labels and their respective four most relevant words (10 topics).

Topic	λ	Assigned label	Words
1	0.6	Public servant remuneration	servants, servant, limitation, remuneration
2	0	Criminal Law	narcotic, hydrometer, clandestine, interrogation
3	0.6	Pension Law	benefit, event, retirement, pension
4	0.6	Civil Law	bank, contract, consumer, *projudi*
5	0.6	Right to health	health, city, municipal, medication
6	0.4	OCR errors	*ento*, no, *ro*, *co*
7	0.6	Tax Law	*icms*, *ipi*, tax, income
8	0	Entities	*econorte*, *rcte*, *pieter*
9	0.4	Labor Law	*fgts*, *pss*, hours, payroll
10	0.6	Document access	original, site, access, report

Regarding the model with 10 topics, the results show that most topics are identified with legal matters routinely discussed by the STF. That being said, topics 6 and 8 were challenging to label. The lawsuits with the highest proportion of these topics were useful in that enterprise.

In the first case, the most representative lawsuits were found to contain a great amount of OCR noise. The most relevant suit, with 99.99957% topic 6 content, contains the following passage: “r cm emoi oit incm m t i o i m cofl inoioem oulfl tofl cmcmh co ffl ffl ffl a z a z ffl o t a o u ffl otoidtoaz d to a i o tn ffl em cmcocoulococm eo cocm [...]”, which is pure gibberish.

While examining topic 8, we discovered that its most representative lawsuits contained a lot of named entities; e.g., from the 15 most frequent words in the suit with most topic 8 content, 8 referred to people or organisations.

The model with 30 topics, as shown in Table 3.2, was also able to identify interpretable topics, many of them directly related to legal matters discussed by the Court. To label each topic, we once again analyse its most relevant words while varying the value of λ. To label the most challenging topics we also examine their most representative lawsuits. Due to the greater number of topics, some of them deal with much more specific matters than in the case of the model with 10 topics. For example, while the model with fewer topics has only one generic topic for Tax Law, the one with 30 topics has four different topics related to different facets of that legal area (topics 3, 25, 27 and 28).

Table 3.2: Topic labels and their respective four most relevant words (30 topics).

Topic	λ	Assigned label	Words
1	0.6	Civil liability	damage, damages, compensation, non-material
2	0.22	Expiration of social security benefit	benefit, expiration, limit, social security (*previdenciário*)
3	0.6	Tax Law	treasury, tax, revenue, taxation
4	0.1	Miscellaneous - Legal vocabulary, enttities and laws	serial number, *pet*, stamp, *itaperuna*
5	0.4	Public servant bonus	bonus, performance, inactive, evaluation
6	0.4	Rural social security	rural, contribution, LEI_8212, pension
7	0.6	Public servant remuneration readjustment	readjustment, servants, remuneration, *urv*
8	0.4	OCR errors	*ento*, no, *ro*, *ffl*
9	0.6	Members of the military	military, servant, servicemen, servants
10	0	Criminal Law	clandestine, *sepetiba*, semi-open, narcotic
11	0.4	Contract law	contract, contracts, fee, accounts
12	0.05	Technical Councils	*confea*, *crea*, agronomy, LEI_6496
13	0.2	Public tender	tender, candidate, notice, openings
14	0.4	Anticipation of remuneration readjustment	*upag*, *pccs*, labor, LEI_8460
15	0.6	Right to health	health, medication (plural), treatment, medication (singular)
16	0.9	Savings account, interest and monetary correction	correction, monetary, savings account, delay
17	0.6	Document access	original, site, acesse, report
18	0.6	labor complaints	*estran*, *tst*, entity, claimant
19	0.4	Miscellaneous - Consumer Law and Bahia (Brazilian state)	consumer, *salvador*, *bahia*, *pdf*
20	0	Entities - names	*lauxen*, *tainá*, *heloise*, *soeli*
21	0.7	Qualification	*num*, normal, internment, *foz*
22	0.5	insurance	insurance, *previd*, institute, *dpu*
23	0.4	Payroll	hours, *fgts*, payroll, overtime
24	0	Miscellaneous - Organisations, charters and non-Portuguese words	*andaterra*, *peixer*, funds, market
25	0.5	Fiscal documents	*ltda*, *ipi*, *nfe*, *icms*
26	0.4	Rio Grande do Sul (Brazilian state)	*sul* , *grande*, *alegre*, *paese*
27	0.4	Income tax	updated, months, *rra*, *irpf*
28	0.2	Tax Law - circulation of goods	compatible, *issqn*, exit, *eireli*
29	0.2	Miscellaneous - Procedure and Paraná (Brazilian state)	*paraná*, *arq*, *curitiba*, *mov*
30	0.4	Payments	*jam*, *vlr*, received, credit

That said, some of the topics have relevant words that do not belong to related matters. Topic 19, for example, assigns high probabilities to words related to both Consumer Law and the Brazilian state of Bahia, with mentions to cities such as Bahia's capital city Salvador. On the other hand, there are topics with very specific relevant words, such as topic 20, that groups names of people. These results can be explained by the nature of the data, which combines various types of documents; e. g. petitions, judgments, orders, proxy statements, certificates, and other supporting documents. We expect that by training only on the Court's rulings the topics would be even more related to specific legal matters discussed by the Justices.

Quantitative analysis of topic distribution as text representation

Figure 3.2 compares the performance on the validation set of classifiers trained on text features obtained from models with 10, 30, 100, 300 and 1,000 topics. All models greatly outperformed a baseline that simply assigns all themes to each instance. Increasing the dimensionality of the representation up to 300 topics improves performance. The model with 1,000 topics, on the other hand, is comparable to the one with 300.

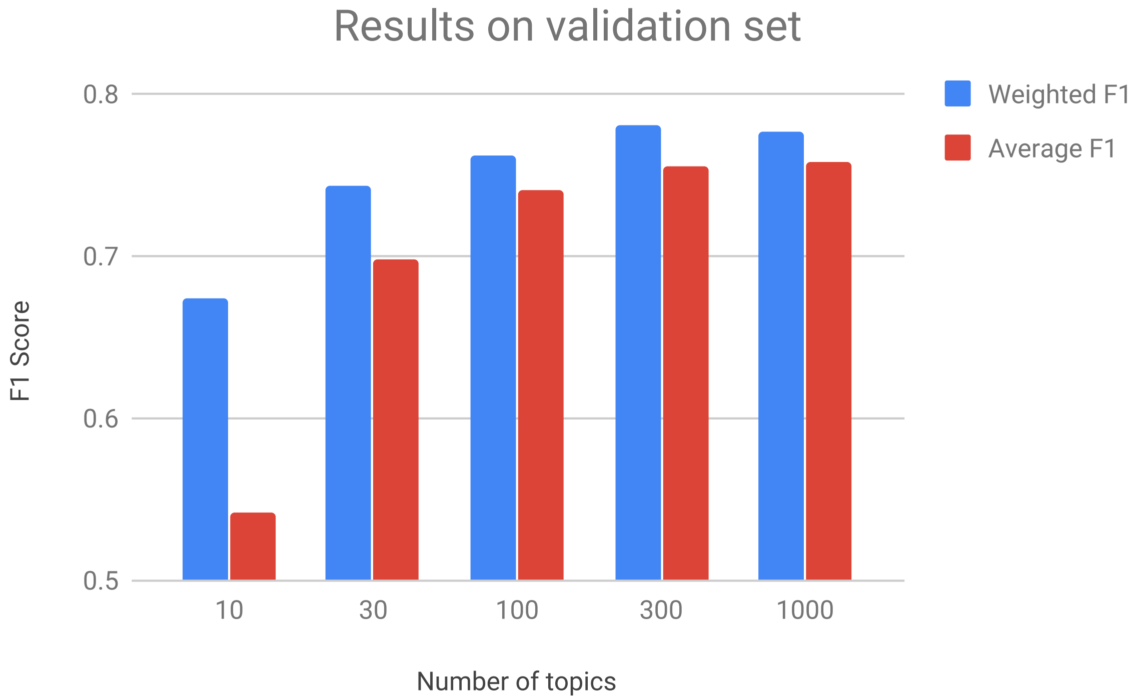


Figure 3.2: Validation set performance of classifiers trained with different numbers of topics.

Table 3.3 compares the 300-dimensional lawsuit representation with the BOW of word counts and tf-idf values representations on the test set. Even though the topic distribution representation enabled good performance, it was not able to outperform the traditional methods. That being said, it has an advantage over the traditional approaches with respect to the dimensionality of the representation—it describes a lawsuit using 300 dimensions instead of 81,418, a relative reduction of 99.63%. As a result, the training and inference is much faster. Furthermore, the smaller number of parameters suggests the topic representation may be better suited to tasks that suffer from lack of data, though more experiments are required to confirm the hypothesis.

Table 3.3: F_1 scores (in %) on the test set of each text representation method. Assigning all themes to all samples yield a weighted F_1 score of 41.17 and an average F_1 score of 5.48.

Theme	Word counts	Tf-idf	300 topics
0	**90.11**	89.63	88.12
5	94.12	**95.81**	93.36
6	68.00	**77.99**	70.79
26	**96.67**	91.53	75.47
33	**82.87**	79.55	67.42
139	86.27	**88.46**	72.00
163	84.35	**86.49**	81.33
232	65.28	**70.67**	52.86
313	70.00	**76.92**	75.93
339	**77.53**	76.29	19.31
350	**83.87**	79.57	82.22
406	84.06	**87.32**	78.26
409	86.79	**87.90**	83.13
555	**80.00**	70.37	50.00
589	**87.80**	86.40	85.94
597	**96.77**	**96.77**	92.86
634	92.72	**95.36**	90.91
660	88.81	**88.87**	52.45
695	**96.65**	**96.65**	96.62
729	95.45	95.45	**97.78**
766	75.61	**82.76**	48.72
773	**96.35**	96.30	94.74
793	89.36	**92.31**	80.00
800	**98.74**	98.41	95.20
810	**94.58**	93.42	83.77
852	84.77	**85.91**	80.00
895	97.33	**97.67**	18.65
951	**99.54**	**99.54**	97.67
975	94.29	**98.55**	92.96
Weighted	**89.29**	89.22	78.07
Average	87.54	**88.37**	75.81

3.1.7 Summary

We proposed the use of LDA to build topic models of Extraordinary Appeals from Brazil's Supreme Court. We labelled and analysed the models with 10 and 30 topics, showing the correspondence between them and legal matters that reach the Court. We compared topic distribution vectors with different number of topics and traditional BOW approaches (tf-idf and word counts) as document representations for a supervised multi-label classification task. The topic distribution representation, with an optimal value of 300 topics, achieved good results using much lower dimensionality than the traditional methods. The technique can be leveraged to help organize, explore and extract information of the massive amounts of data that reach the Court.

3.2 Conclusions

Our results have shown that introducing more topics can be useful if one wishes for topic with finer semantics, that is, with more specific subject matter. On the other hand, a greater number of topics may increase the likelihood of meaningless (OCR artifacts) or jumbled (miscellanea) topics. We have also trained classification models using topic distribution as input and compared them with traditional BOW models. In the next chapter, we will dive into the entity-level to examine Named Entity Recognition.

Part II

Entity-level

Chapter 4

Named Entity Recognition

Named Entity Recognition (NER), the process of locating and classifying named entities in unstructured text, is useful for applications where it is desirable to identify mentions of person names, points in time, organisations, locations, quantities, monetary values, and others, like in systems dealing with the medical or legal fields. Such categories are pre-defined and differ across domain applications; e.g. a NER system for medical documents may include categories for medicine and illness named entities, while a system for processing court orders would probably search for mentions to previous cases.

Although state-of-the-art English NER models are approaching human performance, they do not generalise well to other domains [78]. Research on domain adaptation and transfer learning for NER may help address this issue by creating models that are more robust across different genres and domains and by better leveraging existing annotated corpora. Therefore, the scarcity of publicly available datasets for Named Entity Recognition in languages such as Portuguese motivates the annotation of new corpora in order to support research in that direction.

In Section 4.1 we propose, LeNER-Br, a dataset of manually annotated Brazilian legal documents for Named Entity Recognition. We train LSTM-CRF models on an existing Portuguese NER corpus, achieving better results than previously reported, and on LeNER-Br, creating a benchmark for future methods trained on our data.

4.1 LeNER-Br: a dataset for Named Entity Recognition in Brazilian legal text

Named Entity Recognition (NER) systems have the untapped potential to extract information from legal documents, which can improve information retrieval and decision-making processes. In this section we present a dataset for named entity recognition in

Brazilian legal documents. Unlike other Portuguese language datasets, this dataset is composed entirely of legal documents. In addition to tags for persons, locations, time entities and organisations, the dataset contains specific tags for law and legal cases entities. To establish a set of baseline results, we first performed experiments on another Portuguese dataset: Paramopama [79]. This evaluation demonstrate that LSTM-CRF gives results that are significantly better than those previously reported. We then retrained LSTM-CRF, on our dataset and obtained F_1 scores of 97.04 and 88.82 for Legislation and Legal case token identification, respectively, and F_1 scores of 94.06 and 81.98 when considering only full entity identification of those entities as correct. These results show the viability of the proposed dataset for legal applications.[1]

4.1.1 Introduction

The state-of-the-art entity recognition systems [53, 80] are based on Machine Learning (ML) techniques, employing statistical models that need to be trained on a large amount of labelled data to achieve good performance and generalisation capabilities [81]. The process of labelling data is expensive and time consuming since the best corpora are manually tagged by humans.

There are few manually annotated corpora in Portuguese. Some examples are the first and second HAREM [82, 83] and Paramopama [79]. Another approach is to automatically tag a corpus, like the one proposed in [84] that originated the WikiNER corpus. Such datasets have lower quality than manually tagged ones, as they do not take into consideration sentence context, which can result in inconsistencies between named entity categories [79].

An area that can potentially leverage the information extraction capabilities of NER is the judiciary. The identification and classification of named entities in legal texts, with the inclusion of juridical categories, enable applications such as providing links to cited laws and legal cases and clustering of similar documents.

There are some issues that discourage the use of models trained on existing Portuguese corpora for legal text processing. Foremost, legal documents have some idiosyncrasies regarding capitalization, punctuation and structure. This particularity can be exemplified by the excerpts below:

> EMENTA: APELAÇÃO CÍVEL - AÇÃO DE INDENIZAÇÃO POR DANOS MORAIS - PRELIMINAR - ARGUIDA PELO MINISTÉRIO PÚBLICO EM GRAU RECURSAL - NULIDADE - AUSÊNCIA DE INTERVENÇÃO DO PARQUET NA INSTÂNCIA A QUO - PRESENÇA DE INCAPAZ - PREJUÍZO EXISTENTE - PRELIMINAR ACOLHIDA - NULIDADE RECONHECIDA.

[1] An early version of this section has been published in: Luz de Araujo, P. H. et al. LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text [17].

> HABEAS CORPUS 110.260 SÃO PAULO RELATOR : MIN. LUIZ FUX PACTE.(S) :LAERCIO BRAZ PEREIRA SALES IMPTE.(S) :DEFENSORIA PÚBLICA DA UNIÃO PROC.(A/S)(ES) :DEFENSOR PÚBLICO-GERAL FEDERAL COATOR(A/S)(ES) :SUPERIOR TRIBUNAL DE JUSTIÇA

In these passages, not only are all letters capitalized, but also there is no ordinary phrase structure of subject and predicate. Intuitively, it follows that the distribution of such documents differs from the existing corpora in a way that models trained on them will perform poorly when processing legal documents. Also, as they do not have specific tags for juridical entities, the models would fail to extract such legal knowledge.

This work proposes a Portuguese language dataset for named entity recognition composed entirely of manually annotated legal documents. Furthermore two new categories (LEGISLACAO, for named entities referring to laws; and JURISPRUDENCIA, for named entities referring to legal cases) are added to better extract legal knowledge.

Some efforts have been made on NER in legal texts. For instance, Dozier et al. [32] propose a NER system for Title, Document Type, Jurisdiction, Court and Judge tagging. Nevertheless, only the first entity is identified using a statistical approach, while the others are classified with contextual rules and lookup tables. Cardellino et al. [33] used the Wikipedia to generate an automatically annotated corpus, tagging persons, organisations, documents, abstraction (rights, legal doctrine) and act (statutes) entities. As far as we are aware, we are the first to propose a benchmark dataset and a baseline method for NER in Brazilian legal texts[2].

The rest of this section is organised as follows. First, we discuss the dataset creation process (4.1.2). We then present the model used to evaluate our dataset (4.1.3), along with the training of the model and our choice of hyperparameters (4.1.4). Following that, we present the results achieved regarding the test sets (4.1.5) and our final considerations (4.1.6).

4.1.2 The LeNER-Br dataset

To compose the dataset, 66 legal documents from several Brazilian Courts were collected. Courts of superior and state levels were considered, such as *Supremo Tribunal Federal*, *Superior Tribunal de Justiça*, *Tribunal de Justiça de Minas Gerais* and *Tribunal de Contas da União*. In addition, four legislation documents were collected, such as *Lei Maria da Penha*, resulting in a total of 70 documents.

[2] Resources (data, code and trained model) from this section are available at `https://cic.unb.br/~teodecampos/LeNER-Br/`

For each document, the NLTK [85] library was used to split the text into a list of sentences and tokenize them. The final output for each document is a file with one word per line and an empty line delimiting the end of a sentence.

After preprocessing the documents, WebAnno [86] was employed to manually annotate each one of the documents with the following tags: "ORGANIZACAO" for organisations, "PESSOA" for persons, "TEMPO" for time entities, "LOCAL" for locations, "LEGISLACAO" for laws and "JURISPRUDENCIA" for decisions regarding legal cases. The last two refer to entities that correspond to "Act of Law" and "Decision" classes from the Legal Knowledge Interchange Format ontology [87] respectively.

The IOB tagging scheme [54] was used, where "B-" indicates that a tag is the beginning of a named entity, "I-" indicates that a tag is inside a named entity and "O-" indicates that a token does not pertain to any named entity. Named entities are assumed to be non-overlapping and not spanning more than one sentence.

To create the dataset, 50 documents were randomly sampled for the training set and 10 documents for each of the development and test sets. The total number of tokens in LeNER-Br is comparable to other named entity recognition corpora such as Paramopama and CONLL-2003 English [88] datasets (318,073, 310,000 and 301,418 tokens respectively). Table 4.1 presents the number of tokens and sentences of each set and Table 4.2 displays the number of words in named entities of each set per class. Table 4.3 presents an excerpt from the training set.

Table 4.1: Sentence, token and document count for each set.

Set	Documents	Sentences	Tokens
Training set	50	7,827	229,277
Development set	10	1,176	41,166
Test set	10	1,389	47,630

Table 4.2: Named entity word count for each set.

Category	Training set	Development set	Test set
Person	4,612	894	735
Legal cases	3,967	743	660
Time	2,343	543	260
Location	1,417	244	132
Legislation	13,039	2,609	2,669
Organisation	6,671	1,608	1,367

Table 4.3: Two excerpts from the training set. Each line has a word, a space delimiter and the tag corresponding to the word. Sentences are separated by an empty line.

A	O	TJMG	B-ORGANIZACAO
falta	O	-	O
de	O	Apelação	B-JURISPRUDENCIA
intervenção	O	Cível	I-JURISPRUDENCIA
do	O	1.0549.15.003028-2/003	I-JURISPRUDENCIA
Ministério	B-ORGANIZACAO	,	O
Público	I-ORGANIZACAO	Relator	O
nas	O	(	O
ações	O	a	O
em	O	)	O
que	O	:	O
deva	O	Des	O
figurar	O	.	O
como	O	(	O
fiscal	O	a	O
da	O	)	O
lei	O	Otávio	B-PESSOA
e	O	Portes	I-PESSOA
da	O	,	O
Constituição	B-LEGISLACAO	16ª	B-ORGANIZACAO
(	O	CÂMARA	I-ORGANIZACAO
custus	O	CÍVEL	I-ORGANIZACAO
legis	O	,	O
et	O	julgamento	O
constituitionis	O	em	O
’	O	28/09/2017	B-TEMPO
)	O	,	O
enseja	O	publicação	O
de	O	da	O
forma	O	súmula	O
inexorável	O	em	O
a	O	06/10/2017	B-TEMPO
nulidade	O	)	O
do	O	Assim	O
processo	O	sendo	O
,	O	,	O
segundo	O	entendo	O
prescreve	O	que	O
o	O	deve	O
artigo	B-LEGISLACAO	ser	O
279	I-LEGISLACAO	acolhida	O
...	...	...	...

4.1.3 The baseline model: LSTM-CRF

To establish a methodological baseline on our dataset, we chose the LSTM-CRF model, proposed in [53]. This model is proven to be capable of achieving state-of-the-art performance on the English CoNLL-2003 test set [88] (an F_1 of 90.94). It also has readily available open source implementations [89], which was adapted for the needs of the present work.

The architecture of the model consists of a Bidirectional [49] Long Short-Term Memory (LSTM) [50] followed by a Conditional Random Fields (CRF) [51] layer. The input of the model is a sequence of vector representations of individual words constructed from the concatenation of both word embeddings and character level embeddings.

For the word lookup table we used 300 dimensional GloVe [10] word embeddings pre-trained on a multi-genre corpus formed by both Brazilian and European Portuguese texts [90]. These word embeddings are fine tuned during training.

The character level embeddings are obtained from a character lookup table initialized at random values with embeddings for every character in the dataset. The embeddings are fed to a separate bidirectional LSTM layer. The output is then concatenated with the pre-trained word embeddings, resulting in the final vector representation of the word. Figure 4.1 presents an overview of this process.

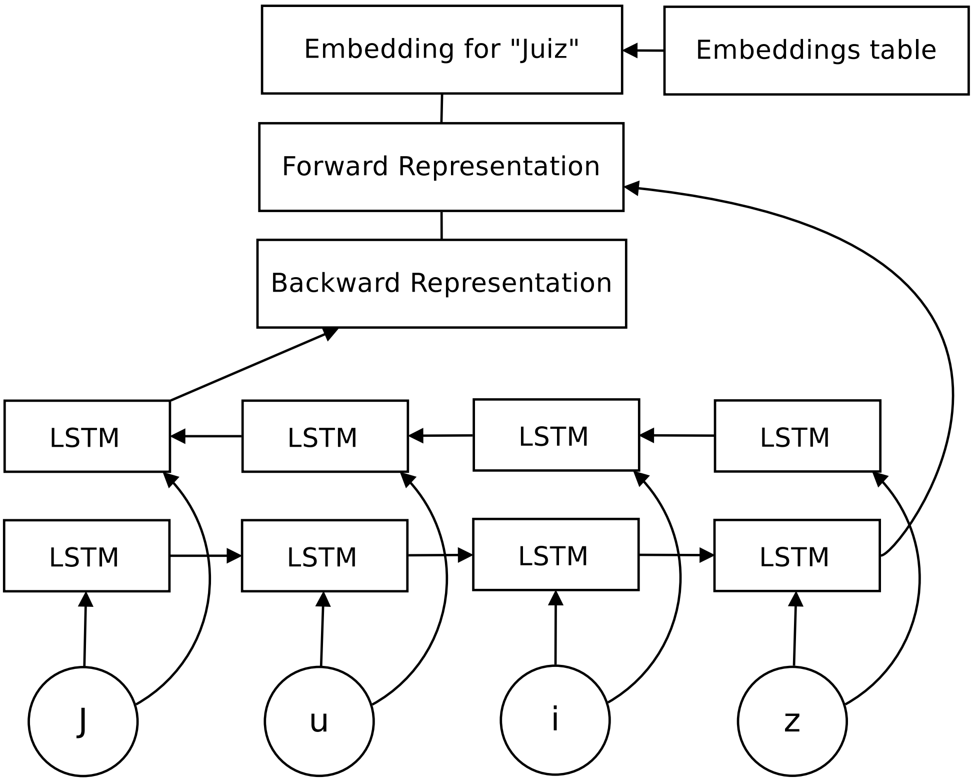


Figure 4.1: Each word vector representation is a result of the concatenation of the outputs of a bidirectional LSTM and the word level representation from the word lookup table.

To reduce overfitting and improve the generalisation capabilities of the model a dropout mask [63] is applied to the outputs of both bidirectional LSTM layers, i.e. the one following the character embeddings and the one after the final word representation. Figure 4.2 shows the main architecture of the model.

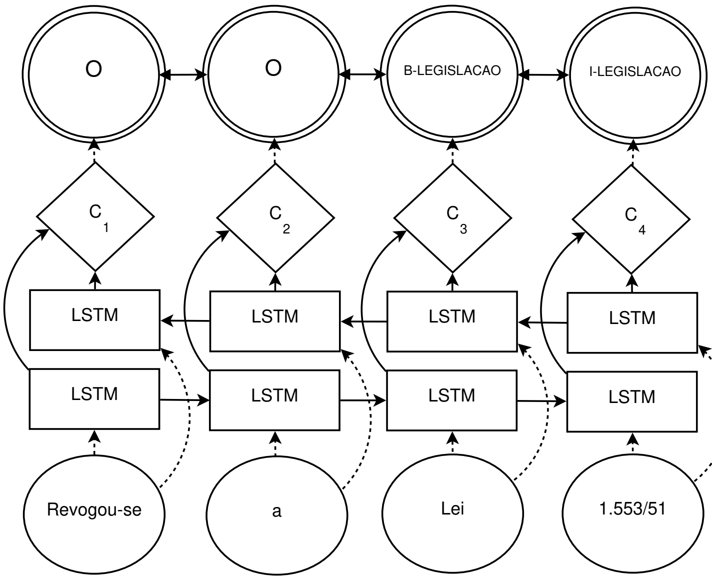


Figure 4.2: The LSTM-CRF model. The word vector representations serve as input to a bidirectional LSTM layer. C_i represents the concatenation of left and right context of word *i*. Dotted lines represent connections after a dropout layer is applied.

4.1.4 Experiments and hyperparameters setting

Here we present the methods employed to train the model and displays the hyperparameters that achieved the best performance.

Both Adam [48] and Stochastic Gradient Descent (SGD) with momentum were evaluated as optmisers. Although SGD had slower convergence, it achieved better scores than Adam. Gradient clipping was employed to prevent the gradients from exploding.

After experimenting with hyperparameters, the best performance was achieved with the ones used in [53], presented in Table 4.4. It is worth noting that the number of LSTM units refers to one direction only. Since the LSTM are bidirectional, the final number of units doubles. Moreover, the learning rate decay is applied after every epoch. The net parameters were saved only when achieving better performance on the validation set than past epochs.

The model was first trained using the Paramopama Corpus [79] to evaluate if it could achieve state-of-the-art performance on a Portuguese dataset. This dataset contains four different named entities: persons, organisations, locations and time entities. After con-

Table 4.4: Model hyperparameter values.

Hyperparameter	Value
Word embedding dimension	300
Character embedding dimension	50
Number of epochs	55
Dropout rate	0.5
Batch size	10
Optmiser	SGD
Learning rate	0.015
Learning rate decay	0.95
Gradient clipping threshold	5
First LSTM layer hidden units	25
Second LSTM layer hidden units	100

firming that the model performed better than the state-of-the-art model (ParamopamaWNN [91]), the LSTM-CRF network was trained with the proposed dataset.

The preprocessing steps applied were lowercasing the words and replacing every digit with a zero. Both steps are necessary to match the preprocessing of the pre-trained word embeddings. Since the character-level representation preserves the capitalization, this information is not lost when the words are lowercased.

4.1.5 Results

The metric used to evaluate the performance of the model on both datasets was the F_1 Score. Tables 4.5 and 4.6 compare the performance of the LSTM-CRF [53] and ParamopamaWNN [91] models on different test sets. Test Set 1 and Test Set 2 are the last 10% of the WikiNER [84] and HAREM [82] corpora respectively. Table 4.7 shows the token prediction scores achieved by the LSTM-CRF model when training on the proposed dataset, that is, correctness is assessed for each token individually. Table 4.8 presents the entity prediction scores, where all tokens in an entity must be assigned to their proper class for it to count as a correct classification. The best precision, recall and F_1 scores for each entity are marked in bold. We do not report results for entity classification when using the Paramopama dataset, since it does not use a tagging scheme that enables the unambiguous identification of entity boundaries.

The obtained results show that the LSTM-CRF network outperforms the ParamopamaWNN on both test sets, achieving better precision, recall and F_1 scores in the majority of the entities. Furthermore, it improved the overall score by 2.48 p.p. and 4.58 p.p. on the first and second test sets respectively.

As far as we are aware, there is no published material about legal entities recognition in Portuguese, so it was not possible to establish a baseline for comparison on LeNER-Br.

Table 4.5: Results (in %) on Paramopama Test Set 1 (10% of the WikiNER [84]) for token classification.

	ParamopamaWNN			LSTM-CRF		
Entity	Precision	Recall	F_1	Precision	Recall	F_1
Person	83.76	90.50	87.00	**91.80**	**92.43**	**92.11**
Location	87.55	**88.09**	87.82	**92.80**	87.39	**90.02**
Organisation	69.55	82.35	75.41	**72.27**	**83.94**	**77.67**
Time	86.96	89.06	88.00	**92.54**	**96.66**	**94.56**
Overall	86.45	89.77	88.08	**90.01**	**91.16**	**90.50**

Table 4.6: Results (in %) on Paramopama Test Set 2 (HAREM [82]) for token classification.

	ParamopamaWNN			LSTM-CRF		
Entity	Precision	Recall	F_1	Precision	Recall	F_1
Person	84.36	88.67	86.46	**94.10**	**95.78**	**94.93**
Location	84.08	86.85	85.44	**90.51**	**92.26**	**91.38**
Organisation	81.48	54.15	65.06	**83.33**	**78.46**	**80.82**
Time	**98.37**	87.40	92.56	91.73	**94.01**	**92.86**
Overall	83.83	88.65	86.17	**90.44**	**91.10**	**90.75**

Table 4.7: Results (in %) on LeNER-Br test set for token classification.

Entity	Precision	Recall	F_1
Person	94.44	92.52	93.47
Location	61.24	59.85	60.54
Organisation	91.27	85.66	88.38
Time	91.15	91.15	91.15
Legislation	97.08	97.00	97.04
Legal cases	87.39	90.30	88.82
Overall	93.21	91.91	92.53

Table 4.8: Results (in %) on LeNER-Br test set for entity classification.

Entity	Precision	Recall	F_1
Person	85.58	78.97	82.14
Location	69.77	63.83	66.67
Organisation	88.30	82.83	85.48
Time	91.30	87.50	89.36
Legislation	93.93	94.18	94.06
Legal cases	79.29	84.86	81.98
Overall	87.98	85.29	86.61

Despite that, the obtained results on LeNER-Br show that a model trained with it can achieve performance in legal cases and legislation recognition comparable to the ones seen in Paramopama entities, with F_1 scores of 88.82% and 97.04% respectively. In addition, person, time entities and organisation classification scores were compatible with the ones observed in the Paramopama scenarios, obtaining scores greater than 80%.

However, location entities have a noticeably lower score than the others on LeNER-Br. This drop could be due to many different reasons. The most important one is probably the fact that words belonging to location entities are rare in LeNER-Br, representing 0.61% and 0.28% of the words pertaining to entities in the train and test sets respectively. Furthermore, location entities are easily mislabelled, as there are words that, depending on the context, may refer to a person, a location or a organisation. A good example is treating the name of an avenue as the name of a person. For instance, instead of identifying "avenida José Faria da Rocha" as a location, the model classifies "José Faria da Rocha" as a person.

4.1.6 Summary

We present LeNER-Br, a Portuguese language dataset for named entity recognition applied to legal documents. As far as we are aware, this is the first dataset of its kind. LeNER-Br consists entirely of manually annotated legislation and legal cases texts and contains tags for persons, locations, time entities, organisations, legislation and legal cases. A state-of-the-art machine learning model, the LSTM-CRF, trained on this dataset was able to achieve a good performance: average F_1 score of 92.53 and 86.81 for token and entity classification, respectively. There is room for improvement, which means that this dataset will be relevant to benchmark methods that are sill to be proposed.

Future work would include the expansion of the dataset, adding legal documents from different courts and other kinds of legislation, e.g. Brazilian Constitution, State Constitutions, Civil and Criminal Codes, among others. In addition, the use of word embeddings pre-trained on a large corpus of legislation and legal documents could potentially improve the performance of the model.

4.2 Conclusions

In this chapter we have proposed a legal domain dataset for NER by manually annotating Brazilian Court documents and legislation. We have trained models using pre-trained word embeddings, LSTM layers as the feature extractor and CRF as a classifier, achieving better results than previously reported on a general domain Brazilian NER corpus and providing a benchmark for future work on our dataset. In the next chapter, we will build upon our examination of entity-level processing to propose work on Entity Linking.

Chapter 5

Proposal for Entity Linking

5.1 Introduction

Entity Linking (EL) goes one step beyond Named Entity Recognition (NER) by linking extracted mentions to entities in a Knowledge Base (KB), such as Wikipedia, specifying exactly which entity is being mentioned. For example, given the sentence *Olympia is the capital of Washington*, an EL system should assign *Washington* to the entity [*Washington (state)*] and not to [*Washington, D.C.*], [*George Washington*] or any other Washington. EL benefits applications where identifying meaningful entities amidst less relevant data is useful, such as in recommender, dialogue and information retrieval systems.

Entity Linking may be performed in three steps:

1. Mention Detection (MD): the system extracts text spans of potential entity mentions—identical to NER in case mentions are restricted to named entities;
2. Candidate Generation (CG): the system assembles a set of entity candidates for each mention; and
3. Entity Disambiguation (ED): the system selects the most probable entity for each mention.

Linkers can perform all three steps or just the last two: the former case is called an end-to-end approach; the latter, disambiguation-only. Formally, given a text document $D = \{w_1, \cdots, w_n\}$, where each w_i is a token from a vocabulary set V, an end-to-end EL model outputs a list of mention-entity pairs where each mention is a span of the input document $m = w_q \cdots w_r$ and each entity is an entry in a KB [92]. In disambiguation-only systems the list of entity mentions is given as an input and the task is simply linking each mention to its corresponding entity in the set of all entities $\mathcal{E} = \{e_i\}_{i=1,\cdots,k}$, where k is the number of entities [93].

The entity set can be massive—possibly reaching millions of entities—which makes the task challenging. Two factors further complicate the problem: mention diversity, as an entity can be represented by different mentions (e.g. *New York*, *NY* and *Big Apple* can all refer to *New York (City)*); and mention ambiguity, as the same mention can represent different entities (e.g. is *Paris* the city or the socialite?).

To solve such problems, EL systems leverage resources like large annotated datasets, structured data and linking statistics. For example, the majority of Wikipedia mentions ($\approx 80\%$ [94]) can be solved by a baseline that, given a mention m, chooses the entity e that maximizes $p(e|m)$. This value is in practice approximated by counting the fraction of times mention m is linked to e in the training set.

A good estimation of this conditional probability requires a large, labelled corpus though, which should not be assumed for low-resource languages or domains as such annotation is expensive and time-consuming. In addition, this feature is bad in the case of rare entities and simply does not work for unseen ones. Thus, a research effort should be directed to developing linkers for domains with scarcity of data and resources.

This chapter aims to propose an EL system for low-resource scenarios, where we do not assume a large labelled target-domain corpus, frequency statistics, canonical text descriptions and structured entity data. The main motivation is the challenges faced in the KnEDLe Project[1], a research effort whose aim is to extract structured information from official publications. One of the tasks of interest is EL—in a scenario of scarcity of resources, such as the one described. As there is no in-domain annotated data for EL yet, we intend to use publicly available corpora (more about that in Section 5.3); but the knowledge acquired thorough our research will be useful and applicable when data is available.

We aim to iterate over the following steps until we are satisfied with the system accuracy (or run out of time):

1. implement an Entity Linking prototype;
2. compare it on established benchmarks with sensible baselines and previous work;
3. analyse the quantitative and qualitative results; and
4. improve the linker.

This chapter is organised as follows. First (5.2), we examine recent research on EL. Then (5.3) we detail what we want to achieve, how we intend to do it, and when we expect to conclude each step.

[1] `https://unb-knedle.github.io/`.

5.2 Related Work

In this section we examine works in the frontier of EL research. We focus on five key aspects concerning features from the techniques studied: two regarding model capabilities—end-to-end linking and global information leveraging—and three related to the assumptions the proposed systems rely on—frequency statistics, structured data and entity dictionary. Table 5.1 summarises our analysis.

Table 5.1: Related work comparison. End-to-End: performs MD—otherwise mention boundaries are assumed. Global: global information. Statistics: entity-mention frequency statistics. Str. Data: structured data. Dictionary: entity dictionary.

		Capabilities		Resources		
Authors	Year	End-to-End	Global	Statistics	Str. Data	Dictionary
Tsai et al. [95]	2016		✓	✓		
Ganea et al. [96]	2017		✓	✓		✓
Pappu et al. [97]	2017	✓	✓	✓		✓
Upadhyay et al. [98]	2018		✓	✓	✓	
Kolitsas et al. [92]	2018	✓	✓	✓		✓*
Gillick et al. [99]	2019				✓	✓
Le et al. [100]	2019				✓	
Logeswaran et al. [93]	2019					✓
Le et al. [101]	2019		✓	✓	✓	✓*
Broscheit [102]	2019	✓				
Wu et al. [103]	2019					✓
Onoe et al. [104]	2020			✓	✓	

*Indirectly: uses entity embeddings trained with entity dictionary.

By **end-to-end linking** we mean systems that not only perform Candidate Generation and Entity Disambiguation but also Mention Detection; otherwise, mention boundaries are assumed to be provided, either by gold annotations or by pre-processing the input with an entity recogniser. Entity linkers that leverage **global information** are those that perform global resolution of mentions; i.e. consider the whole document to perform ED, instead of examining only the local context of each mention.

Large labelled corpora enable analysis of **frequency statistics**, which in turn are used to estimate entity popularity and conditional probabilities of entity given mention [93]. **Structured data** are resources such as relationship information between entities and entity type annotation. Finally, an **entity dictionary** is a set of entities and their respective text description, such as their Wikipedia page, for example.

We now proceed to examine how the listed works reflect each aspect.

5.2.1 End-to-end linking

End-to-end entity linking systems learn[2] and perform all three steps involved in the task. The dependency between the tasks motivates the joint modelling of these steps: Mention Detection errors may irrevocably propagate to the following steps [105, 106], while Mention Detection and Entity Disambiguation can improve one another—greater accuracy for disambiguation promotes better mention boundaries and greater recall for MD enriches the context for disambiguation [92].

Pappu et al. [97] developed a system that performs all three EL steps, albeit in a disconnected manner, as the module for MD was independent. The researchers trained a Named Entity Recognition system for MD by feeding engineered features to a Conditional Random Fields (CRF) classifier. Then they trained entity embeddings and combined them with search click-log data to execute the other two steps.

Kolitsas et al. [92] went one step further in the direction of jointly discovering and linking entities. Their approach considers all possible spans in a text document as potential mentions and learns contextual similarity scores (Ψ) over the entity candidates. A hyperparameter δ is tuned on the validation set so that only potential mention-entity pairs with Ψ score greater than δ are linked—and so MD and ED are performed concurrently.

Broscheit [102] simplified EL to a sequence modelling task that classifies each token over the entire entity vocabulary: in their case, more than 700 thousand categories. Table 5.2 illustrates the approach. Broscheit attached an output classification layer on top of BERT [2] and trained the architecture on Wikipedia text data. Though the method did not outperform the one proposed by Kolitsas et al. [92], it is free from the entity dictionary and frequency statistics assumptions the latter relies on.

5.2.2 Global information

Two types of contextual cues are studied in Entity Disambiguation research: local information, which includes words occurring in a context window around a mention; and global information, which leverages document-level coherence of entities [96]. Local context is used in all studied papers and seems to be essential to the task, since the words surrounding a mention are highly informative of the referred entity. Though global information is less important, it is still helpful, since the mentions present in a document can disambiguate other mentions. For example, the mentions *Seattle*, *Pacific* and *Olympia* suggest the mention *Washington* refers to the state, instead of the president or the city.

Tsai and Roth [95] engineered two features that capture global context: `other-mentions`(m), a set of vectors that represent the other mentions in the document;

[2] CG may not involve learning, as heuristics are commonly used.

Table 5.2: EL as sequence modelling. A Wikipedia link is predicted for each token in a mention, while "O" denotes a Nil prediction. Example reproduced from Broscheit [102].

Text	Label
a	O
deity	Deity
appearing	O
in	O
American	American_comic_book
comic	American_comic_book
book	American_comic_book
s	O
published	O
by	O
Marvel	Marvel_Comics
Comics	Marvel_Comics
.	O
He	O
first	O
appeared	O
in	O
"	O
Thor	Thor_(Marvel_Comics)
"	O

and `previous-titles`(m), a set of vectors that represent the entities in the document that were previously disambiguated. These features (among others) were used to train a linear ranking SVM for ED and greatly improved performance, especially `other-mentions`. The benefit was greater in hard cases, where the correct entity is not the most common one given the mention.

Ganea and Hofmann [96] used CRF to leverage document coherence among entities. The model combines two scoring terms, one for similarity between mention and local context (local information) and one for coherence between an entity and all the others previously mentioned in the document (global information).

Le and Titov [101] combined local context entity-mention similarity scores with pairwise compatibility scores between entities. The latter uses pre-trained entity embeddings and attention weights that measure how relevant each entity is for predicting the others in the document. The researchers perform an ablation analysis that shows: i) local context modelling is essential—dropping it results in a substantial reduction in performance on AIDA CONLL [107] development set (88.05 to 82.41 F_1 score); and ii) global information is beneficial—its elimination results in a 1.2 % drop in performance.

Upadhyay et al. [98] adopted a similar strategy, where the document context d_m of a mention m in a document $\mathcal{D}$ is defined as a bag of all the other mentions in $\mathcal{D}$. A feed-forward layer encodes the document context into a vector $\mathbf{d}$, which is combined to a local context vector and used for ED.

Pappu et al. [97] captured global context when training entity and word embeddings. Each Wikipedia article in the dataset is represented as two sequences of mentioned i) entities and ii) words. When training the entity embeddings, the researchers used each entity to predict their surrounding entities. Consequently, embeddings for coherent entities are clustered together in the projected space.

Kolitsas et al. [92] developed a voting mechanism for global disambiguation. First, a set of mention-entity pairs that are allowed to participate is defined; i.e. those with a local score that surpasses a threshold tuned on the validation set. Then, the final global score for entity candidate e_j of mention m, $G(e_j, m)$, is the cosine similarity between the embedding for e_j and an averaged representation of all voting entities that are other mentions' candidates.

5.2.3 Frequency statistics

When large labelled corpora are available, systems can use mention-entity co-occurrence counts to estimate entity popularity (entity prior or $p(e)$) and the probability of a mention m linking to an entity e (conditional probability of e given m or $p(e|m)$). Such statistics are powerful features for Candidate Generation and Entity Disambiguation and can help construct alias tables of possible mentions for an entity.

Tsai and Roth [95] used frequency statistics for CG. They proposed a two-step approach: i) map a mention string to possible entities by exact matching, sort the candidates by $p(e|m)$ and return the top k candidates; if the first step fails to generate any candidate, ii) break the mention into its tokens w_i, map them to entities through partial matching and rank the candidates by $p(e|w_i)$. They also used the conditional probability as a feature for disambiguation. In fact, most works [96, 98, 92, 101] employed $p(e|m)$ both for CG and as a feature for ED.

Pappu et al. [97] estimated $p(e|m)$ by making use of anonymized search engine data that links user queries to Wikipedia pages. For example, *Barack* and *President Obama* map to *wiki/Barack_Obama.* Onoe and Durrett [104] used $p(e|m)$ for CG and as a backup plan for entities with few annotated types, where their entity type prediction approach would fail to precisely disambiguate.

5.2.4 Structured data

Relationship tuples and entity type annotation can be used to improve ED [93]. One example is including the fine-grained types of mentions to help linkers choose entities of the appropriate type: if the mention *Washington* has the gold type `states_of_the_west_coast`, disambiguation to the entity *George_Washington_(President)* is discouraged. The same can be said in the case of relationship tuples: a linker having access to the tuple *(Barack Obama, Spouse, Michelle Obama)* can more easily link the mention *Michelle* to the correct entity when *Barack Obama* is also present in the document.

Upadhyay et al. [98] included type information in their EL system by using their mention context vector to predict the set of the fine-grained types of the mention in addition to its referred entity. The researchers assumed the types to be the same for both mention and linked entity. The results show that adding such structured knowledge improves accuracy when compared to the system with no type prediction training.

Gillick et al. [99] used Wikipedia categories as one of the sources of information for entity encoding. When T-SNE [108] projects the obtained entity vectors to a two-dimensional space, entities of the same type are clustered together even in the case of high word overlap with entities of different types: *Montreal (city)* is not close to *Of Montreal (band)* but to *Beirut (City)*—the learned embeddings are fundamentally different from standard word embeddings.

Le and Titov [100] trained embeddings for types and combined them to compute entity vectors. Let $\mathbf{t}$ be the vector for type t, and T_e the set of all types of entity e. Then the vector for e is

$$\mathbf{e} = \mathrm{ReLU}\left(\mathbf{W}_e \frac{1}{|T_e|} \sum_{t \in T_c} \mathbf{t} + \mathbf{b}_e\right) , \tag{5.1}$$

where $\mathbf{W}_e$ is a weight matrix and $\mathbf{b}_e$ is a bias vector. The obtained embeddings are used to score compatibility between context-mention pair and entities.

Le and Titov [101] used Wikipedia link data to better re-rank candidate lists. They constructed an undirected graph where the vertices are the entities in the KB. Vertices e_u and e_v are connected if there is a document D_{wiki} such that: i) D_{wiki} in an article describing e_u and e_v is mentioned in it; or ii) both entities are present in the document and there are less than l entities between them. The graph is then used to penalise candidate entity assignments that contain unlinked pairs.

Claiming that neural models tend to overfit by memorizing properties of the most frequent entities in a dataset, Onoe and Durrett [104] changed the EL task focus: instead of directly predicting entities given mentions, they modelled the fine-grained entity properties. The intuition is that the proposed approach can better disambiguate closely related entities and generalise. Their system consists of a learned entity typing model and an untrained entity link predictor based on the type predictions. The approach greatly outperforms baselines on a test set of unseen mentions during training (62.2% accuracy versus a second best of 54.1%).

5.2.5 Entity dictionary

Most works we studied assumed the existence of an entity dictionary $\mathcal{E} = \{(e_i, d_i)\}_{i=1,...,k}$ for training EL systems, where d_i is a text description of entity e_i and k is the number of entities. The text description data is commonly compared with the mention context in order to aid ED.

Ganea and Hofmann [96] collected word-entity co-occurrence counts, $\#(w, e)$, from: i) the entity canonical text description (its Wikipedia article in their case); and ii) words surrounding mentions to the entity. These counts were used to generate a "positive" distribution of words related to the entity $\hat{p}(w|e) \propto \#(w, e)$, in contrast to $q(w)$, a generic word probability distribution to sample negative—unrelated to the entity—words. The authors used the distributions and a max-margin objective to infer entity embeddings such that vectors of positive words are closer to it than vectors of random words.

Pappu et al. [97] pre-processed Wikipedia articles by transforming hyperlinks to entities into their article title (canonical form). Each article a is then represented as: i) the sequence of entities it mentions $(e_1, \cdots, e_n)$; and ii) the sequence of tokens it contains $(w_1, \cdots, w_m)$. The data was used to create a d-dimensional representation of tokens and entities in a common vector space.

Gillick et al. [99] also assumed an entity dictionary: one of their main sources of information for their proposed entity encoder is the first paragraph of the entity Wikipedia article. The paragraph enconder consists in averaging the unigram and bigram embeddings and feeding the two vectors to a Fully-Connected (FC) layer. The output is combined with a categories vector and a title vector to compute the final entity encoding.

Logeswaran et al. [93] and Wu et al. [103] both employed BERT [2] to assess compatibility between a context-mention pair and an entity. Given a mention m, its left and right context c_l and c_r, an entity e, and the entity description d, the input to the transformer is

$$[\texttt{CLS}]\ \mathrm{c}_l\ [\texttt{M}_s]\ m\ [\texttt{M}_e]\ \mathrm{c}_r\ [\texttt{SEP}]\ e\ [\texttt{ENT}]\ d\ [\texttt{SEP}],$$

where [CLS], [$\texttt{M}_s$], [$\texttt{M}_e$] and [SEP] are special tokens: the context-candidate embedding is given by last layer of the output of [CLS]; [$\texttt{M}_s$] and [$\texttt{M}_e$] tag mention boundaries; [SEP] is a BERT separator token; and [ENT] separates entity title and description. This construction enables the transformer to jointly attend to context and entity description. Wu et al. use a similar approach to perform CG by modelling entity and mention-in-context separately using a bi-encoder.

Kolitsas et al. [92] and Le et al. [101] indirectly assumed an entity dictionary since they borrowed the entity embeddings trained by Ganea and Hofmann [96]. Both works compute similarity between mentions and entities by combining the entity vector, the computed probability $p(e|m)$ and the mention context encoded by a LSTM network, and feeding them to a FC layer.

5.3 Work plan

The scenario that assumes resources such as structured data, entity dictionary and large labelled corpora is not realistic in the case of low-resource languages and domains with incipient KBs (medical or legal fields, for example). Thus, strategies should be explored to develop linking methods that rely on weaker assumptions.

We plan to develop an EL system for such scenarios, establishing three main desiderata[3]:

1. independence from entity dictionary;
2. independence from frequency statistics; and
3. independence from structured data.

These features would enable the proposed system to be able to work in the cases where the KB consists simply of entity IDs without text descriptions.

5.3.1 Modelling

Broscheit's work [102] is the only one we examined that follows all of the desiderata. But the simplification made—reducing entity linking to a sequence tagging task—introduces one serious issue. Since the classes (entities) are fixed, the whole model must be retrained every time new entities are introduced to the knowledge base. This is not feasible: training just one epoch takes between one and three days on two Nvidia TitanXp/1080Ti GPUs. That said, one possible line of investigation is fine-tuning the learned parameters to other domains and entity sets.

Transfer learning can be particularly helpful when target labelled data is not so abundant. Thus, we intend to leverage large labelled datasets by pre-training on such corpora and fine-tuning and evaluating on low-resource domains. This is similar to previous work on zero-shot EL [93, 103], where the scientists used a model pre-trained on large corpora [2] and then fine-tuned it on the zero-shot dataset introduced by Logeswaran et al. [93]. One major problem we will face is how to model entity vectors—those works assumed entity dictionaries; we do not. Possible baselines are training entity embeddings [96] or feeding an entity and the most common words found near its mentions to a transformer [93, 103].

[3] Due to the already challenging nature of the problem, we leave the desirable traits of training end-to-end and leveraging global information to future work.

Alternatively, we can treat the task as a distance learning problem, where we build a model that learns a vector space where the euclidean distance corresponds to mention-entity similarity. We can do that by minimising a triplet loss objective [109]. Originally proposed for face recognition, the triplet loss penalises distance between an anchor and a positive—in our case an entity-mention pair—and encourages distance between the anchor and a negative—the entity and a unrelated mention. It is defined as:

$$L = \sum_{i=1}^{n} \max(\|f(x_i^a) - f(x_i^p)\|^2 - \|f(x_i^a) - f(x_i^n)\|^2 + \alpha, 0)\,, \tag{5.2}$$

where $f(\cdot)$ is a function representing the encoder, x_i^a is an anchor (in our case an entity), x_i^p is a positive example (a mention to the entity), x_i^n is a negative example (an unrelated mention), n is the number of training triplets, and α is a margin to be enforced between negative and positive pairs.

We are aware of one work [110] that uses the triplet loss for EL. The researchers applied the triplet loss to rank entity candidates in the medical domain. There is a lot of room for improvement though: the work used a shallow CNN as the encoder, only mention and entity spans were used as input, and word2vec [11] and fasttext [12] were used as pre-trained embeddings. The use of more recent advances—transformer encoders that are aware of local context and leverage contextual embeddings—should be investigated.

We also plan to examine other SOTA methods for EL[4] to build a more comprehensive overview of existing approaches.

5.3.2 Datasets

In this subsection we introduce some corpora with EL annotation.

Wikipedia Wikipedia is widely used for EL training and evaluation: the articles titles can be used as entities, the article body as their text description, and the hyperlinks' anchor texts as mentions. The May 2019 Wikipedia dump used by Wu et al. [103] contains 9 million mentions and 5.9 million entities.

Wikia zero-shot corpus The zero-shot EL dataset proposed by Logeswaran et al. [93] contains documents from 16 Wikias ranging from various domains, such as American Football, Doctor Who and World of Warcraft. Eight of the Wikias are used for training, four for validation and four for testing. In addition, the validation and test sets do not contain entities seen during training. To make the task more challenging, the mentions

[4] A compilation of EL state-of-the-art methods can be found in `http://nlpprogress.com/english/entity_linking.html`.

that can be linked to the correct entity by simple string matching are downsampled to occupy only 5% of the final dataset, which contains 49,275 labeled mentions for training, and 10,000 for validation and testing each. The entity sets for each Wikia range from 10,000 to 100,000 entities. This dataset has two main desirable traits for our work: it's smaller than Wikipedia, which is more adequate to our desired low-resource scenario; and, as the testing and validation splits contain only unseen entities, we can evaluate how well the system adapts to an expanding entity set, which is mostly always the case in real life applications.

TACKBP-2010 The TACKBP-2010 [111] is a established benchmark for EL systems. The dataset is composed of news and web documents with mention-entity pair annotation. The entities set is composed of 818,741 entities from the TAC Reference KB.

We plan to examine other datasets, such as the AIDA CONLL-Yago dataset [107], the original WikilinksNED dataset [112], and the Unseen-Mentions version created by Onoe and Durret [104].

5.3.3 Evaluation

For evaluation, we plan to report the metrics commonly adopted by EL works:

Recall@k Recall@k measures performance of the Candidate Generation task. It is the fraction of generated candidate lists that contain the correct entity among the top-k candidates. That is, given a total of m candidate lists of size k, if n of them contain the correct entity, $n \leq m$, then

$$\text{Recall@k} = \frac{n}{m}\,. \tag{5.3}$$

This metric represents the upper-bound of Entity Disambiguation performance: a system cannot possibly select the correct entity if it is not in the set of candidates.

Unnormalised accuracy The unnormalised accuracy is the fraction of mentions that were assigned to the correct entity, computed on the entire test set. Given a total of m mentions, if c of them are linked to the correct entity, then

$$\text{Unnormalised accuracy} = \frac{c}{m}\,. \tag{5.4}$$

The best value for the unnormalised accuracy is the Recall@k. Higher is better.

Normalised accuracy The normalised accuracy computes the above metric considering only the subset of mentions whose correct entity is among the retrieved top-k candidates. Given a total of n mentions whose correct entity is covered by the generated candidate list, if d of them are linked to the correct entity, then

$$\text{Normalised accuracy} = \frac{d}{n} \tag{5.5}$$

The best value for the normalised accuracy is 1. Higher is better.

5.3.4 Schedule

We divide the remaining activities in four groups: Study, Experiments, Writing and Wrap Up. Figure 5.1 summarises the planned schedule for the research.

Study will comprise two months, comprehending mainly reading activities. The first month will be dedicated to researching low-resource and zero-shot Entity Linking in order to investigate previous approaches, knowledge gaps and possible research directions. The second month will be dedicated to research on EL datasets and baselines in order to identify corpora and baseline to work and compare with.

We allocate three months to execute the Experiments: developing a baseline that works reasonably well (one month) and then concentrating efforts on improving the linker (two months). The first month overlaps with the dataset and baseline research task, as a way to combine theory and practice. Choosing an architecture, training the parameters, tuning the hyper-parameters, evaluating the trained model and comparing with previous work are activities for the other two months. Model development will be realized iteratively, following a cycle of: i) training, ii) evaluating, iii) (hopefully) improving the model and iv) starting a new cycle.

Two months will be dedicated to writing the dissertation, that is, a background, an EL and a conclusion chapters. The first month overlaps with the model development task; this should not be an issue as the planned tasks are independent. The milestone of submitting the work to the defense board concludes the group.

Finally, the last month will be dedicated to wrapping up. We will submit the findings to conferences/journals and prepare the dissertation presentation, which will conclude the group.

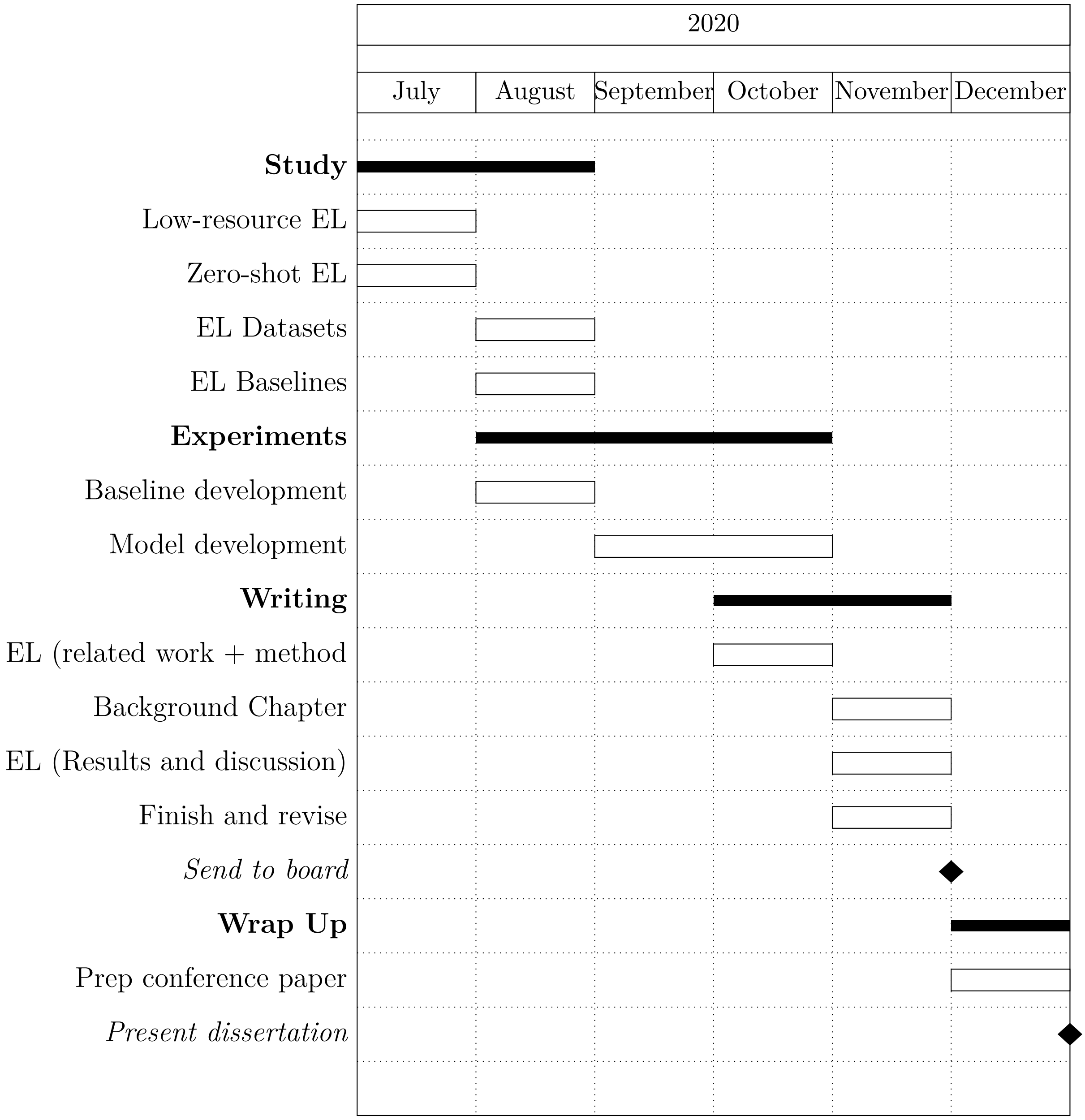


Figure 5.1: Monthly plan of attack.

References

[1] Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le: *XLNet: Generalized Autoregressive Pretraining for Language Understanding.* In Wallach, H., H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (editors): *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc., 2019. `http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf`. 1

[2] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* CoRR, abs/1810.04805, 2018. `http://arxiv.org/abs/1810.04805`. 1, 23, 61, 65, 66

[3] Howard, Jeremy and Sebastian Ruder: *Fine-tuned Language Models for Text Classification.* CoRR, abs/1801.06146, 2018. `http://arxiv.org/abs/1801.06146`. 1, 22, 23, 27, 29, 31

[4] Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi: *Findings of the 2017 Conference on Machine Translation (WMT17).* In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. `https://www.aclweb.org/anthology/W17-4717`. 1

[5] Wu, Felix, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli: *Pay Less Attention with Lightweight and Dynamic Convolutions.* CoRR, abs/1901.10430, 2019. `http://arxiv.org/abs/1901.10430`. 1

[6] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin: *Attention is All you Need.* In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (editors): *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`. 1

[7] Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever: *Improving language understanding by generative pre-training*, 2018. Available at `https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf`. 1

[8] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov: *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* CoRR, abs/1907.11692, 2019. `http://arxiv.org/abs/1907.11692`. 1

[9] Ruder, Sebastian: *Neural Transfer Learning for Natural Language Processing.* PhD thesis, National University of Ireland, Galway, 2019. 1

[10] Pennington, Jeffrey, Richard Socher, and Christopher Manning: *GloVe: Global vectors for word representation.* In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 1, 52

[11] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean: *Distributed Representations of Words and Phrases and their Compositionality.* In Burges, C. J. C., L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (editors): *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. `http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf`. 1, 67

[12] Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov: *Enriching Word Vectors with Subword Information.* arXiv preprint arXiv:1607.04606, 2016. 1, 67

[13] Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer: *Deep Contextualized Word Representations.* In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. `https://www.aclweb.org/anthology/N18-1202`. 1

[14] Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever: *Language Models are Unsupervised Multitask Learners.* OpenAI blog, 1(8), February 2019. `https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`. 1, 23

[15] Araujo, Pedro Henrique Luz de, Teófilo Emídio de Campos, Fabricio Ataides Braz, and Nilton Correia da Silva: *VICTOR: a Dataset for Brazilian Legal Documents Classification.* In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1449–1458, Marseille, France, May 2020. European Language Resources Association, ISBN 979-10-95546-34-4. `https://www.aclweb.org/anthology/2020.lrec-1.181`. 3, 6, 37

[16] Luz de Araujo, Pedro H., Teófilo E. de Campos, and Marcelo Magalhaes Silva de Sousa: *Inferring the source official texts: can SVM beat ULMFiT?* In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), Evora, Portugal, March 2-4

2020. Springer. https://propor.di.uevora.pt/, Code and data available from https://cic.unb.br/~teodecampos/KnEDLe/. 3

[17] Luz de Araujo, Pedro H., Teófilo E. de Campos, Renato R. R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo: *LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text.* In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Canela, RS, Brazil, September 24-26 2018. 3, 8, 23, 35, 37, 48

[18] Zhang, Xiang, Junbo Zhao, and Yann LeCun: *Character-level Convolutional Networks for Text Classification.* In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS, pages 649–657, Cambridge, MA, USA, 2015. MIT Press. https://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.pdf. 5, 8

[19] Deshpande, V. P., R. F. Erbacher, and C. Harris: *An Evaluation of Naïve Bayesian Anti-Spam Filtering Techniques.* In *IEEE SMC Information Assurance and Security Workshop*, pages 333–340, June 2007. 5

[20] Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau: *Sentiment Analysis of Twitter Data.* In *Proceedings of the Workshop on Languages in Social Media*, page 30–38, USA, 2011. Association for Computational Linguistics, ISBN 9781932432961. 5

[21] Wang, Canhui, Min Zhang, Shaoping Ma, and Liyun Ru: *Automatic Online News Issue Construction in Web Environment.* In *Proceedings of the 17th International Conference on World Wide Web*, page 457–466, New York, NY, USA, 2008. Association for Computing Machinery, ISBN 9781605580852. https://doi.org/10.1145/1367497.1367560. 5

[22] Ruder, Sebastian: *Neural Transfer Learning for Natural Language Processing.* PhD thesis, National University of Ireland, Galway, 2019. 5

[23] Cássia Carvalho Lopes, Rita de: *Eventual Influences of Common Law on the Brazilian Legal SysBrazilian Legal System*, Mars 2017. https://www.migalhas.com/HotTopics/63,MI255372,51045-Eventual+Influences+of+Common+Law+on+the+Brazilian+Legal+System, [Online; posted 15-Mars-2017. https://www.migalhas.com/HotTopics/63,MI255372,51045-Eventual+Influences+of+Common+Law+on+the+Brazilian+Legal+System]. 6, 35

[24] Fariello, Luiza: *CNJ apresenta Justiça em Números 2018, com dados dos 90 tribunais*, August 2018. http://www.cnj.jus.br/noticias/cnj/87512-cnj-apresenta-justica-em-numeros-2018-com-dados-dos-90-tribunais, [Online; posted 27-August-2018. http://www.cnj.jus.br/noticias/cnj/87512-cnj-apresenta-justica-em-numeros-2018-com-dados-dos-90-tribunais]. 6

[25] Secretaria de Comunicação Social do Conselho Nacional de Justiça: *Sumário Executivo do Relatório Justiça em Números 2018*, 2018. http://www.cnj.jus.br/files/conteudo/arquivo/2018/09/da64a36ddee693ddf735b9ec03319e84.pdf. 6

[26] Joachims, Thorsten: *Text categorization with Support Vector Machines: Learning with many relevant features.* In Nédellec, Claire and Céline Rouveirol (editors): *Machine Learning: ECML*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg, ISBN 978-3-540-69781-7. 7

[27] Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov: *Bag of Tricks for Efficient Text Classification.* In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, 2017. `http://aclweb.org/anthology/E17-2068`. 7

[28] Liu, C., W. Hsaio, C. Lee, T. Chang, and T. Kuo: *Semi-Supervised Text Classification With Universum Learning.* IEEE Transactions on Cybernetics, 46(2):462–473, Feb 2016, ISSN 2168-2267. 8

[29] Conneau, Alexis, Holger Schwenk, Loïc Barrault, and Yann Lecun: *Very Deep Convolutional Networks for Text Classification.* In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain, April 2017. Association for Computational Linguistics. `http://www.aclweb.org/anthology/E17-1104`. 8, 13

[30] Johnson, Rie and Tong Zhang: *Supervised and Semi-supervised Text Categorization Using LSTM for Region Embeddings.* In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML, pages 526–534. JMLR.org, 2016. `http://proceedings.mlr.press/v48/johnson16.pdf`. 8

[31] Howard, Jeremy and Sebastian Ruder: *Universal Language Model Fine-tuning for Text Classification.* In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. `http://www.aclweb.org/anthology/P18-1031`. 8

[32] Dozier, Christopher, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali: *Named entity recognition and resolution in legal text.* In *Semantic Processing of Legal Texts*, pages 27–43. Springer, 2010. 8, 23, 37, 49

[33] Cardellino, Cristian, Milagro Teruel, Laura Alonso Alemany, and Serena Villata: *A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker.* In *Proceedints of the 16th International Conference on Artificial Intelligence and Law (ICAIL)*, London, United Kingdom, June 2017. Preprint available from `https://hal.archives-ouvertes.fr/hal-01541446`. 8, 23, 37, 49

[34] Kanapala, Ambedkar, Sukomal Pal, and Rajendra Pamula: *Text summarization from legal documents: a survey.* Artificial Intelligence Review, Jun 2017, ISSN 1573-7462. `https://doi.org/10.1007/s10462-017-9566-2`. 8, 23, 37

[35] Galgani, Filippo, Paul Compton, and Achim Hoffmann: *Combining Different Summarization Techniques for Legal Text.* In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, HYBRID, pages 115–123, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. `https://www.aclweb.org/anthology/W12-0515.pdf`. 8, 23, 37

[36] Kumar, Ravi and K Raghuveer: *Legal document summarization using latent dirichlet allocation.* International Journal of Computer Science and Telecommunications, 3:114–117, 2012. 8, 23, 37

[37] Kim, Mi Young, Ying Xu, and Randy Goebel: *Summarization of Legal Texts with High Cohesion and Automatic Compression Rate.* In *New frontiers in artificial intelligence.* Springer, 2013. 8, 23, 37

[38] Carter, David J, James Brown, and Adel Rahmani: *Reading the high court at a distance: Topic modelling the legal subject matter and judicial activity of the high court of Australia, 1903-2015.* UNSWLJ, 39:1300, 2016. 8, 37, 39

[39] Remmits, Ylja: *Finding the Topics of Case Law: Latent Dirichlet Allocation on Supreme Court Decisions*, 2017. Bachelor's thesis, Radboud University, July 2017. 8, 37, 39

[40] O'Neill, J, C Robin, L O'Brien, and P Buitelaar: *An analysis of topic modelling for legislative texts.* In *Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts*, June 2016. 8, 37, 39

[41] Katz, Daniel Martin, II Bommarito, Michael J, and Josh Blackman: *Predicting the Behavior of the Supreme Court of the United States: A General Approach.* arXiv e-prints, page arXiv:1407.6333, Jul 2014. 8, 23, 37

[42] Aletras, Nikolaos, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos: *Predicting Judicial Decisions of the European Court of Human Rights: A Natural Language Processing Perspective.* PeerJ in Computer Science, October 2016. 8, 23, 37

[43] Şulea, Octavia Maria, Marcos Zampieri, Mihaela Vela, and Josef van Genabith: *Predicting the Law Area and Decisions of French Supreme Court Cases.* In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP*, pages 716–722. INCOMA Ltd., 2017. `https://doi.org/10.26615/978-954-452-049-6_092`. 8, 23, 37

[44] Undavia, S., A. Meyers, and J. E. Ortega: *A Comparative Study of Classifying Legal Documents with Neural Networks.* In *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 515–522, Sep. 2018. 8, 37

[45] Smith, Ray: *An overview of the Tesseract OCR engine.* In *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, volume 2, pages 629–633. IEEE, 2007. 9

[46] Braz, Fabricio Ataides, Nilton Correia da Silva, Teofilo Emidio de Campos, Felipe Borges S. Chaves, Marcelo H. S. Ferreira, Pedro Henrique Inazawa, Victor H. D. Coelho, Bernardo Pablo Sukiennik, Ana Paula Goncalves Soares de Almeida, Flavio Barros Vidal, Davi Alves Bezerra, Davi B. Gusmao, Gabriel G. Ziegler, Ricardo V. C. Fernandes, Roberta Zumblick, and Fabiano Hartmann Peixoto: *Document classification using a Bi-LSTM to unclog Brazil's supreme court.* In *NeurIPS workshop on Machine Learning for the Developing World (ML4D)*, December 8 2018. Event webpage: `https://sites.google.com/view/ml4d-nips-2018/`. Published at arXiv:1811.11569. 9

[47] Silva, N. Correia da, F. A. Braz, T. E. de Campos, D.B. Gusmao, F.B. Chaves, D.B. Mendes, D.A. Bezerra, G.G. Ziegler, L.H. Horinouchi, M.H.P. Ferreira, G.H.T.A. Carvalho, R. V. C. Fernandes, F. H. Peixoto, M. S. Maia Filho, B. P. Sukiennik, L. S. Rosa, R. Z. M. Silva, and T. A. Junquilho: *Document type classification for Brazil's supreme court using a Convolutional Neural Network.* In *10th International Conference on Forensic Computer Science and Cyber Law (ICoFCS)*, Sao Paulo, Brazil, October 29-30 2018. Winner of the best paper award. 9, 23, 35, 37

[48] Kingma, Diederik P and Jimmy Ba: *Adam: A method for stochastic optmisation.* In *International Conference on Learning Representations (ICLR)*, 2015. Preprint available at `https://arxiv.org/abs/1412.6980`. 14, 53

[49] Graves, Alex and Jürgen Schmidhuber: *Framewise phoneme classification with bidirectional LSTM and other neural network architectures.* Neural Networks, 18(5-6):602–610, 2005. 14, 52

[50] Hochreiter, Sepp and Jürgen Schmidhuber: *Long short-term memory.* Neural computation, 9(8):1735–1780, 1997. 14, 52

[51] Lafferty, John D., McCallum andrew, and Fernando C. N. Pereira: *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.* In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc., ISBN 1-55860-778-1. `https://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers`. 15, 52

[52] Huang, Zhiheng, Wei Xu, and Kai Yu: *Bidirectional LSTM-CRF Models for Sequence Tagging.* CoRR, abs/1508.01991, 2015. `http://arxiv.org/abs/1508.01991`. 15

[53] Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer: *Neural Architectures for Named Entity Recognition.* In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. `https://www.aclweb.org/anthology/N16-1030`. 15, 48, 52, 53, 54

[54] Ramshaw, Lance A. and Mitchell P. Marcus: *Text chunking using transformation-based learning.* In *Natural language processing using very large corpora*, pages

157–176. Springer, 1999. Preprint available at `http://arxiv.org/abs/cmp-lg/9505040`. 15, 50

[55] Chen, Tianqi and Carlos Guestrin: *XGBoost: A Scalable Tree Boosting System.* CoRR, abs/1603.02754, 2016. `http://arxiv.org/abs/1603.02754`. 18, 33, 40

[56] Luz de Araujo, Pedro H., Teófilo E. de Campos, and Marcelo Magalhaes Silva de Sousa: *Inferring the source official texts: can SVM beat ULMFiT?* In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), Evora, Portugal, March 2-4 2020. Springer. `https://propor.di.uevora.pt/`, Code and data available from `https://cic.unb.br/~teodecampos/KnEDLe/`. 22

[57] Vargas Feijó, Diego de and Viviane Pereira Moreira: *RulingBR: A Summarization Dataset for Legal Texts.* In Villavicencio, Aline, Viviane Moreira, Alberto Abad, Helena Caseli, Pablo Gamallo, Carlos Ramisch, Hugo Gonçalo Oliveira, and Gustavo Henrique Paetzold (editors): *Computational Processing of the Portuguese Language*, pages 255–264, Cham, 2018. Springer International Publishing, ISBN 978-3-319-99722-3. 23, 35, 37

[58] Kudo, Taku and John Richardson: *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, November 2018. Association for Computational Linguistics (ACL). 26

[59] Hearst, Marti A.: *Support Vector Machines.* IEEE Intelligent Systems, 13(4):18–28, July 1998, ISSN 1541-1672. 26

[60] Bradbury, James, Stephen Merity, Caiming Xiong, and Richard Socher: *Quasi-Recurrent Neural Networks.* CoRR, abs/1611.01576, 2016. `http://arxiv.org/abs/1611.01576`. 27

[61] Smith, Leslie N. and Nicholay Topin: *Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates.* CoRR, abs/1708.07120, 2017. `http://arxiv.org/abs/1708.07120`. 27

[62] Ioffe, Sergey and Christian Szegedy: *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, 2015. `http://proceedings.mlr.press/v37/ioffe15.html`. 27

[63] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov: *Dropout: A Simple Way to Prevent Neural Networks from Overfitting.* J. Mach. Learn. Res., 15(1):1929–1958, January 2014, ISSN 1532-4435. `http://jmlr.org/papers/v15/srivastava14a.html`. 27, 53

[64] Nair, Vinod and Geoffrey E. Hinton: *Rectified Linear Units Improve Restricted Boltzmann Machines.* In *Proceedings of the 27th International Conference on Machine Learning (ICLR)*, pages 807–814, USA, 2010. Omnipress,

ISBN 978-1-60558-907-7. https://icml.cc/Conferences/2010/papers/432.pdf. 27

[65] Smith, Leslie N.: *No More Pesky Learning Rate Guessing Games.* CoRR, abs/1506.01186, 2015. http://arxiv.org/abs/1506.01186. 28

[66] Kingma, Diederick P and Jimmy Ba: *Adam: A method for stochastic optmisation.* In *International Conference on Learning Representations (ICLR)*, 2015. 28

[67] Loshchilov, Ilya and Frank Hutter: *Fixing Weight Decay Regularization in Adam.* CoRR, abs/1711.05101, 2017. http://arxiv.org/abs/1711.05101. 28

[68] Blei, David M.: *Probabilistic Topic Models.* Commun. ACM, 55(4):77–84, April 2012, ISSN 0001-0782. http://doi.acm.org/10.1145/2133806.2133826. 35, 36

[69] Mauá, Denis Deratani: *Modelos de tópicos na classificação automática de resenhas de usuários.* Master's thesis, Escola Politécnica da Universidade de São Paulo, 2009. 35

[70] Rubin, Timothy N., America Chambers, Padhraic Smyth, and Mark Steyvers: *Statistical topic models for multi-label document classification.* Machine Learning, 88(1):157–208, Jul 2012, ISSN 1573-0565. https://doi.org/10.1007/s10994-011-5272-5. 35

[71] Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman: *Indexing by latent semantic analysis.* Journal of The American Society for Information Science, 41(6):391–407, 1990. 36

[72] Hofmann, Thomas: *Probabilistic Latent Semantic Indexing.* In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pages 50–57, New York, NY, USA, 1999. ACM, ISBN 1-58113-096-1. http://doi.acm.org/10.1145/312624.312649. 36

[73] Blei, David M., Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation.* Journal of Machine Learning Research, 3:993–1022, March 2003, ISSN 1532-4435. http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf. 36, 39

[74] Blei, David M. and John D. Lafferty: *Dynamic Topic Models.* In *Proceedings of the 23rd International Conference on Machine Learning*, ICML, pages 113–120, New York, NY, USA, 2006. ACM, ISBN 1-59593-383-2. http://doi.acm.org/10.1145/1143844.1143859. 36

[75] Hoffman, Matthew D., David M. Blei, and Francis Bach: *Online Learning for Latent Dirichlet Allocation.* In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS, pages 856–864, USA, 2010. Curran Associates Inc. https://papers.nips.cc/paper/3902-online-learning-for-latent-dirichlet-allocation. 40

[76] Grimmer, Justin and Brandon M. Stewart: *Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.* Political Analysis, 21(3):267–297, 2013. 41

[77] Sievert, Carson and Kenneth Shirley: *LDAvis: A method for visualizing and interpreting topics.* In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. `https://www.aclweb.org/anthology/W14-3110`. 41

[78] Augenstein, Isabelle, Leon Derczynski, and Kalina Bontcheva: *Generalisation in Named Entity Recognition.* Computer Speech and Language, 44(C):61–83, July 2017, ISSN 0885-2308. `https://doi.org/10.1016/j.csl.2017.01.012`. 47

[79] Mendonça Jr., Carlos A. E., Hendrik Macedo, Thiago Bispo, Flávio Santos, Nayara Silva, and Luciano Barbosa: *Paramopama: a Brazilian-Portuguese Corpus for Named Entity Recognition.* In *XII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC).* SBC, 2015. 48, 53

[80] Ma, Xuezhe and Eduard Hovy: *End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF.* In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1064–1074, Berlin, Germany, August 7-12 2016. ACL. Preprint available at `https://arxiv.org/abs/1603.01354`. 48

[81] Mansouri, Alireza, Lilly Suriani Affendey, and Ali Mamat: *Named entity recognition approaches.* International Journal of Computer Science and Network Security, 8(2):339–344, 2008. 48

[82] Santos, Diana and Nuno Cardoso: *A golden resource for named entity recognition in Portuguese.* In *International Workshop on Computational Processing of the Portuguese Language*, pages 69–79. Springer, 2006. 48, 54, 55

[83] Freitas, Cláudia, Cristina Mota, Diana Santos, Hugo Gonçalo Oliveira, and Paula Carvalho: *Second HAREM: Advancing the State of the Art of Named Entity Recognition in Portuguese.* In *Language Resources and Evaluation Conference (LREC).* European Language Resources Association, 2010. 48

[84] Nothman, Joel, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran: *Learning multilingual named entity recognition from Wikipedia.* Artificial Intelligence, 194:151–175, 2013. 48, 54, 55

[85] Bird, Steven, Ewan Klein, and Edward Loper: *Natural language processing with Python: analyzing text with the natural language toolkit.* O'Reilly Media, Inc., 2009. 50

[86] Castilho, Richard Eckart de, Eva Mujdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann: *A web-based tool for the integrated annotation of semantic and syntactic structures.* In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, 2016. 50

[87] Hoekstra, Rinke, Joost Breuker, Marcello Di Bello, and Alexander Boer: *The LKIF Core Ontology of Basic Legal Concepts.* In *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques*, 2007. 50

[88] Tjong Kim Sang, Erik F and Fien De Meulder: *Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition.* In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, volume 4, pages 142–147. Association for Computational Linguistics, 2003. 50, 52

[89] Genthial, Guillaume: *Sequence Tagging - Named Entity Recognition with Tensorflow.* GitHub repository `https://github.com/guillaumegenthial/sequence_tagging/tree/0048d604f7a4e15037875593b331e1268ad6e887`, 2017. 52

[90] Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jessica Rodrigues, and Sandra Aluisio: *Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks.* In *Proceedings of Symposium in Information and Human Language Technology*, Uberlandia, MG, Brazil, October 2–5 2017. Sociedade Brasileira de Computação. Preprint available at `https://arxiv.org/abs/1708.06025`. 52

[91] Mendonça Jr., Carlos A. E. M., Luciano A. Barbosa, and Hendrik T. Macedo: *Uma Arquitetura Híbrida LSTM-CNN para Reconhecimento de Entidades Nomeadas em Textos Naturais em Língua Portuguesa.* In *XIII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC).* SBC, 2016. 54

[92] Kolitsas, Nikolaos, Octavian Eugen Ganea, and Thomas Hofmann: *End-to-End Neural Entity Linking.* In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium, October 2018. Association for Computational Linguistics. `https://www.aclweb.org/anthology/K18-1050`. 58, 60, 61, 63, 65

[93] Logeswaran, Lajanugen, Ming Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee: *Zero-Shot Entity Linking by Reading Entity Descriptions.* In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy, July 2019. Association for Computational Linguistics. `https://www.aclweb.org/anthology/P19-1335`. 58, 60, 63, 65, 66, 67

[94] Ratinov, Lev, Dan Roth, Doug Downey, and Mike Anderson: *Local and Global Algorithms for Disambiguation to Wikipedia.* In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. `https://www.aclweb.org/anthology/P11-1138`. 59

[95] Tsai, Chen Tse and Dan Roth: *Cross-lingual Wikification Using Multilingual Embeddings.* In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598, San Diego, California, June 2016. Association for Computational Linguistics. `https://www.aclweb.org/anthology/N16-1072`. 60, 61, 63

[96] Ganea, Octavian Eugen and Thomas Hofmann: *Deep Joint Entity Disambiguation with Local Neural Attention.* In *Proceedings of the Conference on Empirical Methods*

in Natural Language Processing, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. https://www.aclweb.org/anthology/D17-1277. 60, 61, 62, 63, 65, 66

[97] Pappu, Aasish, Roi Blanco, Yashar Mehdad, Amanda Stent, and Kapil Thadani: *Lightweight Multilingual Entity Extraction and Linking.* In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM, page 365–374, New York, NY, USA, 2017. Association for Computing Machinery, ISBN 9781450346757. https://doi.org/10.1145/3018661.3018724. 60, 61, 63, 65

[98] Upadhyay, Shyam, Nitish Gupta, and Dan Roth: *Joint Multilingual Supervision for Cross-lingual Entity Linking.* In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2486–2495, Brussels, Belgium, 10-11 2018. Association for Computational Linguistics. https://www.aclweb.org/anthology/D18-1270. 60, 62, 63, 64

[99] Gillick, Daniel, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano: *Learning Dense Representations for Entity Retrieval*, 2019. 60, 64, 65

[100] Le, Phong and Ivan Titov: *Distant Learning for Entity Linking with Automatic Noise Detection.* In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4081–4090, Florence, Italy, July 2019. Association for Computational Linguistics. https://www.aclweb.org/anthology/P19-1400. 60, 64

[101] Le, Phong and Ivan Titov: *Boosting Entity Linking Performance by Leveraging Unlabeled Documents.* In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945, Florence, Italy, July 2019. Association for Computational Linguistics. https://www.aclweb.org/anthology/P19-1187. 60, 62, 63, 64, 65

[102] Broscheit, Samuel: *Investigating Entity Knowledge in BERT with Simple Neural End-To-End Entity Linking.* In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China, November 2019. Association for Computational Linguistics. https://www.aclweb.org/anthology/K19-1063. 60, 61, 62, 66

[103] Wu, Ledell, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer: *Zero-shot Entity Linking with Dense Entity Retrieval.* arXiv e-prints, page arXiv:1911.03814, November 2019. 60, 65, 66, 67

[104] Onoe, Yasumasa and Greg Durrett: *Fine-Grained Entity Typing for Domain Independent Entity Linking.* arXiv e-prints, page arXiv:1909.05780, September 2019. 60, 63, 64, 68

[105] Sil, Avirup and Alexander Yates: *Re-Ranking for Joint Named-Entity Recognition and Linking.* In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM, page 2369–2374, New York, NY,

USA, 2013. Association for Computing Machinery, ISBN 9781450322638. https://doi.org/10.1145/2505515.2505601. 61

[106] Luo, Gang, Xiaojiang Huang, Chin Yew Lin, and Zaiqing Nie: *Joint Entity Recognition and Disambiguation.* In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal, September 2015. Association for Computational Linguistics. https://www.aclweb.org/anthology/D15-1104. 61

[107] Hoffart, Johannes, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum: *Robust Disambiguation of Named Entities in Text.* In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. https://www.aclweb.org/anthology/D11-1072. 62, 68

[108] Maaten, Laurens van der and Geoffrey Hinton: *Visualizing Data using t-SNE.* Journal of Machine Learning Research, 9:2579–2605, 2008. http://www.jmlr.org/papers/v9/vandermaaten08a.html. 64

[109] Schroff, Florian, Dmitry Kalenichenko, and James Philbin: *FaceNet: A Unified Embedding for Face Recognition and Clustering.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 67

[110] Mondal, Ishani, Sukannya Purkayastha, Sudeshna Sarkar, Pawan Goyal, Jitesh Pillai, Amitava Bhattacharyya, and Mahanandeeshwar Gattu: *Medical Entity Linking using Triplet Network.* In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 95–100, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. https://www.aclweb.org/anthology/W19-1912. 67

[111] Ji, Heng, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis: *Overview of the TAC 2010 knowledge base population track.* In *Text Analysis Conference*, volume 3, 2010. 68

[112] Eshel, Yotam, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy: *Named Entity Disambiguation for Noisy Text.* In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 58–68, Vancouver, Canada, August 2017. Association for Computational Linguistics. https://www.aclweb.org/anthology/K17-1008. 68