University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Domain-specific datasets for document classification and named entity recognition

Pedro Henrique Luz de Araujo

Dissertation presented for conclusion of the Master Degree in Computer Science

Supervisor
Dr. Teófilo E. de Campos

Brasilia
2021

University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Domain-specific datasets for document classification and named entity recognition

Pedro Henrique Luz de Araujo

Dissertation presented for conclusion of the Master Degree in Computer Science

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

Dr. Alexandre Rademaker
IBM Research and FGV

Dr. Thiago de Paulo Faleiros
CIC/UnB

Dr. Genaina Nunes Rodrigues
Computer Science Graduate Program Coordinator

Brasilia, 2nd August 2021

Acknowledgments

Writing the list of people who helped me producing this document is not quite nerve-wrecking; but it is a little anxiety-inducing, as I fear I might be forgetting someone. Nevertheless, I think it is important to show gratitude to such individuals so I will try anyway. Any omission is not due to a lack of appreciation but to some unfortunate brain glitch.

I'd like to thank my supervisor Teófilo de Campos for guiding me and encouraging me throughout my whole research career, ever since I was a computer engineering freshman.

I'm also thankful for my colleagues, specially Fred Guth, for stimulating exchanges of ideas and discussions, and Patricia Medyna, who faced with me some challenging courses.

I want to thank Cláudia Nalon for her excellent theory of computation course and for lending me interesting books (one of which I still need to return!).

I thank everybody from the Victor and KnEDLe projects, without whom the work I present here would not exist. I send special thanks to Fabricio Braz for his *vai que interessa!* paper notification system.

I also send thanks to my examiners Alexandre Rademaker and Thiago Faleiros for spending their valuable time reading and evaluating my work. Without them this would be a much poorer dissertation. I'm also thankful for the anonymous researchers that peer-reviewed my paper submissions—my work is all the much better after their suggestions.

I thank my loved ones for keeping me healthy, sane and happy.

I'm also grateful for the administrative staff of the Computer Science Department for their efficiency and helpfulness when I had to battle bureaucratic problems.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. We acknowledge the support of "Projeto de Pesquisa & Desenvolvimento de aprendizado de máquina (machine learning) sobre dados judiciais das repercussões gerais do Supremo Tribunal Federal - STF". We are also grateful for the support from Fundação de Apoio à Pesquisa do Distrito Federal (FAPDF, project KnEDLe, convênio 07/2019) and Fundação de Empreendimentos Científicos e Tecnológicos (Finatec). The present work was also supported by CAPES' *Portal de Periódicos*.

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.

Though natural language processing (NLP) has seen a great deal of progress in the last decade, current models require a large number of annotated examples and tend to not generalise beyond training data and domain. Recent transfer learning approaches can mitigate those needs, but specific-domain labelled datasets are still needed to fine-tune pre-trained models and for evaluation.

In this work, we propose three domain-specific datasets with annotated data for two NLP tasks: document classification and named entity recognition (NER). To establish a benchmark for future work on the legal and public administration domains, for each dataset we train, evaluate and compare different models.

First, we propose a dataset for NER in legal documents with domain specific entities and train a biLSTM-CRF model on the data. Next, we propose a dataset of documents from Brazil's Supreme Court annotated with labels for two classification tasks; we train and compare shallow, deep and multimodal models trained on the data with and without sequence modelling; and evaluate topics inferred through latent Dirichlet allocation. Finally, we propose a dataset of official gazette texts with labelled and unlabelled data and compare traditional bag-of-words models trained with linear classifiers with a state-of-the-art transfer learning method (ULMFiT).

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

Resumo expandido

Título: Conjuntos de dados de domínio específico para classificação de documento e reconhecimento de entidade nomeada.

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, propomos três bases de dado de domínio específico com anotação para duas tarefas de PLN: classificação de documento e reconhecimento de entidade nomeada (REN). Para estabelecer uma base de comparação para trabalhos futuros nos domínios de textos jurídicos e da administração pública, para cada conjunto de dados treinamos, avaliamos e comparamos diferentes modelos.

Sistemas de REN têm o potencial de extrair conhecimento de documentos jurídicos e obter insumos que podem melhorar a recuperação de informações e subsidiar tomadas de decisão. Com isso em vista, o primeiro conjunto de dados que apresentamos, o LeNER-Br, trata da tarefa de REN em textos jurídicos brasileiros. Diferentemente de outros conjuntos de dados de textos em português, o LeNER-Br é composto inteiramente de textos jurídicos, mais especificamente, acórdãos, instrumentos normativos e leis. Além de rótulos para entidades genéricas (pessoa, local, organização e tempo), o conjunto de dados conta com anotações para entidades específicas do domínio: legislação e jurisprudência. Para estabelecer resultados de classificação como base para comparações com trabalhos futuros, usamos uma arquitetura biLSTM-CRF para treinar um modelos nos dados e avaliar os resultados. Primeiramente, para testar a viabilidade do método em textos em português, realizamos experimentos na base de REN Paramopama, atingindo resultados

que superaram o estado da arte. Feito isso, retreinamos o modelos no LeNER-Br, onde obtivemos escores F_1 de 97,04 e 88,82 para classificação de token de legislação e jurisprudência, respectivamente, e escores de 94,06 e 81,98 quando somente a identificação exata da entidade é considerada correta.

Nosso segundo conjunto de dados é o VICTOR, composto por documentos digitalizados do Supremo Tribunal Federal (STF). A base reúne mais de 40 mil recursos extraordinários, totalizando cerca de 692 mil documentos, ou 4,6 milhões de páginas. Os dados contêm anotações para duas tarefas: classificação de tipo de documento e identificação de tema de repercussão geral. A primeira trata de classificação por página, em que cada uma pode pertencer a seis classes disjuntas; a segunda trata de classificação por processo e é multi-rótulo: cada processo pode ter mais de um tema de repercussão geral. Para gerar resultados como referência para trabalhos futuros, treinamos uma série de modelos nos dados: modelos de saco-de-palavras, redes neurais convolucionais e recorrentes e *gradient boosted trees*. Também avaliamos a possibilidade de aproveitar a natureza sequencial dos dados para melhorar os resultados de classificação de tipo de documento; para tanto, treinamos um campo aleatório condicional de cadeias lineares nas predições de uma rede convolucional treinada nos dados, método que trouxe melhorias. Finalmente, comparamos um modelo de identificação de tema que utiliza conhecimento específico do domínio para filtrar páginas menos informativas com um modelo regular que utiliza todas as páginas. Ao contrário das expectativas dos especialistas da Corte, constatou-se que é melhor utilizar todas as páginas.

Ainda em relação ao conjunto VICTOR, utilizamos alocação latente de Dirichlet para modelar os recursos extraordinários como uma possível medida pra auxiliar na organização dos casos do STF. Avaliamos a qualidade dos tópicos obtidos de duas maneiras: qualitativamente, a partir da análise das palavras mais relevantes de cada tópico, e quantitativamente, utilizando os vetores de distribuição de tópico como entrada para um classificador de tema de repercussão geral. Inicialmente treinamos modelos de 10 e 30 tópicos para a avaliação qualitativa, ocasião em que identificamos que os tópicos encontrados guardavam relação com matérias de direito. Ficou evidenciado, ainda, a existência de uma tensão entre granularidade e qualidade de tópicos: o modelo de 30 tópicos era capaz de detectar tópicos mais específicos, mas também gerava tópicos que misturavam assuntos distintos. Para a avaliação quantitativa, treinamos modelos adicionais com 100, 300 e 1.000 tópicos, que utilizamos como vetores de características para treinar o classificador de temas. Ao se comparar os resultados obtidos com aqueles resultantes de técnicas de representação de texto tradicionais (saco-de-palavras com contagem de palavras e valores tf-idf), verificou-se que os tópicos, embora não superassem as técnicas tradicionais, conseguiam resultados de classificação aceitáveis, fortalecendo a hipótese de que os tópicos encontra-

dos são relevantes para a administração dos processos. O modelo com 300 tópicos atingiu a melhor performance, conseguindo resultados bons com representações interpretáveis de baixa dimensão.

Como último trabalho na base VICTOR, realizamos um estudo com o objetivo de aproveitar as informações visuais dos documentos para melhorar a classificação de tipo de documento. Para tanto, estendemos a versão pequena do VICTOR para incluir as imagens das páginas, guardadas em formato JPEG. Além disso, retomamos a exploração da modelagem sequencial das páginas como fonte de melhoria de resultados de classificação. Primeiramente, treinamos modelos unimodais de classificação de texto e imagem de maneira independente. Como classificador de imagem, utilizamos um modelo ResNet pré-treinado na base ImageNet e fizemos seu ajuste-fino nas imagens do VICTOR. Como classificador de texto, treinamos uma rede neural convolucional com filtros de tamanhos diferentes nos textos do VICTOR. Uma vez treinados os modelos, usamo-los como extratores de características visuais e textuais, as quais são combinadas por um Módulo de Fusão. Tal módulo consegue lidar com modalidades de entrada faltantes por meio de *embeddings* aprendíveis. As métricas de classificação obtidas pelo modelo de fusão superaram aquelas dos modelos unimodais. Para extração de informações sequenciais, realizamos experimentos com redes biLSTM e campos aleatórios condicionais de cadeias lineares. Os modelos multimodais sequenciais superaram aqueles sem informação de sequência, sendo que o melhor método realizava conjuntamente o aprendizado sequencial e de fusão de informações visuais e textuais.

Finalmente, propomos um conjunto de dados composto por textos do Diário Oficial do Distrito Federal. A motivação de dá por conta de os diários oficiais serem uma rica fonte de informações relevantes para a sociedade—um exame cuidadoso desse tipo de documento pode acarretar a detecção de fraudes e irregularidades e prevenir o mau uso de recursos públicos. Os dados contém tantos textos com anotação de órgão público de origem quanto textos não rotulados. Treinamos, avaliamos e comparamos um modelo estado-da-arte que usa transferência de aprendizado, o ULMFiT, com modelos tradicionais de saco-de-palavras usando Naïve Bayes e SVM como classificadores. O modelo tradicional treinado com SVM mostrou-se competitivo: superou o ULMFiT na métrica de escore F_1 médio, apresentando escore F_1 ponderado e acurácia ligeiramente abaixo aos de seu oponente. Além disso, seu treino e inferência são bem mais rápidos que os do ULMFiT, por conta do menor custo computacional.

Os trabalhos descritos resultaram nas seguintes publicações:

- Luz de Araujo, P. H. et al. LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text [87].

- Luz de Araujo, P. H. et al. VICTOR: a dataset for Brazilian legal documents classification. [86]
- Luz de Araujo, P. H. et al. Topic Modelling Brazilian Supreme Court Lawsuits [85].
- Luz de Araujo, P. H. et al. Inferring the source of official texts: can SVM beat ULMFiT? [88].

Além das principais contribuições deste trabalho—os conjuntos de dado—inferimos de nossos experimentos as seguintes conclusões, as quais consideramos contribuições empíricas:

- Um modelo biLSTM-CRF treinado no dados do LeNER-Br é capaz de reconhecer entidades específicas do domínio jurídico com um grau de acerto equivalente ao do reconhecimentos de entidades genéricas sem necessidade de pré-processamento específico ou engenharia de características.
- Modelos de saco-de-palavras podem atingir resultados de classificação competitivos com os de modelo de aprendizado profundo, especialmente em cenários com menor abundância de dados, como nos casos do Small VICTOR e dos documentos do Diário Oficial do DF.
- Tópicos detectados pelo algoritmo de alocação latente de Dirichlet podem ser usados como um ponto de partida para auxiliar a administração de casos do STF.
- Os resultados de classificação de tipo de documento do STF melhorou com cada modalidade de entrada adicional.

Treinamos modelos com o objetivo de servir de base de apoio para trabalhos futuros. Dado isso e nossos recursos computacionais limitados, não realizamos buscas extensivas por melhores hiper-parâmetros ao treinar redes neurais. Outra limitação do nosso trabalho é o fato de que nossas anotações não contam com métricas de medidas de concordância entre anotadores. Isso se deu por conta de limitações de recursos humanos, de modo que cada documento não foi anotado por mais de uma pessoa. Nos casos dos documentos do LeNER-BR e do Diário Oficial do DF, buscou-se reforçar a correição e consistência da anotação por meio da cuidadosa revisão de todas as anotações. No caso do STF, uma vez que as anotações foram realizadas por servidores do STF durante a execução do fluxo ordinário de trabalho da Corte, não estamos ciente dos detalhes do processo de anotação.

Como trabalho futuros, sugerimos rodar experimentos adicionais com busca abrangente de hiper-parâmetros para verificar modelos de aprendizado profundo podem alcançar melhorias que justifiquem seu alto custo computacional. Seria igualmente interessante o treino ponta-a-ponta do método de aprendizado sequencial multi-modal que propusemos

para os documentos do VICTOR. Por fim, esperamos que nossos dados sejam usados em trabalhos futuros de transferência de aprendizado, adaptação e generalização de domínio e aprendizado multilíngue.

Palavras-chave: processamento de linguagem natural, processamento da língua portuguesa, classificação de texto, modelos de tópicos, reconhecimento de entidade nomeada, classificação multi-modal, transferência de aprendizado

Contents

List of Acronyms and Abbreviations

ARE *Agravo de Recurso Extraordinário*

BERT Bidirectional Encoder Representations from Transformers

BOW Bag-Of-Words

BVic Big VICTOR

CG Candidate Generation

CNN Convolutional Neural Network

CRF Conditional Random Fields

DNN Deep Neural Networks

DODF *Diário Oficial do Distrito Federal*

ED Entity Disambiguation

EL Entity Linking

FC Fully connected

GPT-2 Generative Pre-trained Transformer 2

KB Knowledge Base

KnEDLe Knowledge Extraction from Documents of Legal content

LDA Latent Dirichlet Allocation

LeNER-Br Legal Named Entity Recognition-Brazil

LSI Latent Semantic Indexing

LSTM Long Short-Term Memory

MD Mention Detection

ML Machine Learning

MVic Medium VICTOR

NB Naïve Bayes

NER Named Entity Recognition

NLP Natural Language Processing

OCR Optical Character Recognition

PDF Probability Density Function

PLSI Probabilistic Latent Semantic Indexing

PMF Probability Mass Function

QRNN Quasi-Recurrent Neural Network

RE *Recurso Extraordinário*

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SGD Stochastic Gradient Descent

STF *Supremo Tribunal Federal*

SVD Singular Value Decomposition

SVic Smal VICTOR

SVM Support Vector Machine

tf-idf term frequency-inverse document frequency

ULMFiT Universal Language Model Fine-Tuning

XGBoost eXtreme Gradient Boosting

Notation

Bernoulli	The Bernoulli distribution
Beta	The Beta distribution
Bin	The Binomial distribution
Cov	Covariance
$\boldsymbol{\Sigma}$	Covariance Matrix
Dir	The Dirichlet distribution
$\mathbb{E}$	Expectation
$\mathcal{E}$	Entity Set
Γ	The Gamma function
J	The cost function
η	The learning rate
μ	Mean
Multinomial	The Multinomial distribution
$\boldsymbol{\mu}$	Mean vector
$\mathbb{N}$	The natural numbers
nll	The negative log likelihood loss function
ω	An experiment outcome
P	Probability of an event
$\boldsymbol{\theta}$	Parameter vector
Poisson	The Poisson distribution
$\mathbb{R}$	set of Real numbers
ReLU	The ReLU function
Ω	Sample Space
σ	Standard deviation
sigmoid	The sigmoid function
softmax	The softmax function
tanh	The hyperbolic tangent function
U	The Uniform distribution
$\mathcal{V}$	Vocabulary size

Var	Variance
$\mathbb{Z}$	The integers

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 [149, 31, 57], machine translation [11, 146, 143] and natural language inference [108, 149, 81]. Unfortunately, in addition to requiring a large number of annotated examples, deep NLP models tend to not generalise beyond training data and domain [118]. 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 [105, 96, 10] 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 [106, 31, 57, 109], 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.

To promote research on under-explored domains, we[1] present three novel domain-specific datasets with annotated data for natural language processing tasks: one for named entity recognition; the others, for document classification. In addition, we train and evaluate baseline and state-of-the-art models on each resource to establish benchmarks for comparison. This work was executed in the context of two research projects: Project VICTOR[2], which aimed to partially automatise case management for the Brazilian Supreme Court; and Knowledge Extraction from Documents of Legal content (KnEDLe)[3], whose objective is to extract structured information from the Federal District's official publications in order to facilitate information retrieval.

1.1 Objectives

The main objectives of this dissertation are proposing three domain-specific datasets and creating benchmarks for each of them. These datasets have shared characteristics that justify their joint presence in this dissertation. All of them:

- concern natural language processing tasks;
- are in Portuguese language;
- relate to a specific domain—either legal or official publication documents.

That said, the research effort for each dataset is self-contained and independent of the others, so I dedicate a chapter for each of them. For the same reason, I describe the objectives and hypotheses specific to each work at the start of the corresponding chapter.

1.2 Contributions

Our main contributions are the following datasets:

- LeNER-Br, a dataset of legal documents for named entity recognition with annotation for domain-specific entities.
- VICTOR, a multimodal dataset of legal documents from Brazil's Supreme Court with document type and lawsuit theme annotation.

[1] Though this dissertation has only one author, the work was done in collaboration with others, who are credited in the corresponding chapters. For this reason, I feel is more appropriate to use the pronoun "we" when I was not the only one involved.

[2] `https://ailab.unb.br/projetos/victor`.

[3] `http://nido.unb.br/index.html`.

- a dataset of labelled and unlabelled texts from the Official Gazette of Brazil's Federal District with annotation for document source classification.

In addition to these, we make the following empirical contributions:

- we train and evaluate a named entity recogniser on the LeNER-Br and Paramopama [94] datasets, establishing a benchmark for the former and pushing the state of the art of the latter.
- we create a benchmark for the VICTOR data that compares shallow and deep models trained for each of two goals: document type classification and lawsuit theme assignment.
- we propose a method for sequence-aware multimodal page classification, which we train and evaluate on the VICTOR data.
- we find the topics that occur in the VICTOR data and evaluate their quality by manually inspecting them to analyse their semantics and using topic distribution as a feature for supervised lawsuit classification.
- we train, evaluate and compare bag-of-word models to a language model pre-training based approach using the DODF data.

Our work has generated the following publications:

- Luz de Araujo, P. H. et al. LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text [87].
- Luz de Araujo, P. H. et al. VICTOR: a dataset for Brazilian legal documents classification. [86]
- Luz de Araujo, P. H. et al. Topic Modelling Brazilian Supreme Court Lawsuits [85].
- Luz de Araujo, P. H. et al. Inferring the source of official texts: can SVM beat ULMFiT? [88].
- Luz de Araujo, P. H. et al. Sequence-aware multimodal page classification of Brazilian legal documents. Submitted to the International Journal on Document Analysis and Recognition.

All of our code and data is available in the following page: `https://cic.unb.br/~teodecampos/peluz/`.

1.3 Outline

In Chapter 2 I briefly discuss the concepts used in the rest of the dissertation—mainly regarding probability theory, machine learning, neural networks and natural language processing.

In Chapter 3 we propose a NER dataset of manually annotated legal texts, describe how we trained an entity recogniser on it, and analyse the obtained results.

In Chapter 4 we propose a dataset of legal documents from Brazil's Supreme Court with annotation for two document classification tasks. Since the data is composed from both visual and textual data, we present a method for multimodal document page classification. Moreover, we extract corpus topics through latent Dirichlet allocation, which we quantitatively and qualitatively evaluate by examining topic semantics and using topic distribution vectors as a feature for supervised learning.

In Chapter 5 we propose a dataset of texts from the Official Gazette of The Federal District of Brazil, with both unlabelled and labelled samples for a classification task. We compare a baseline approach that uses bag-of-words modelling with a state-of-the art approach based on unsupervised language model pre-training.

In Chapter 6 we summarise our findings and conclude our work.

In Appendix A we present a research proposal whose aim is to investigate entity linking for low-resource domains.

Chapter 2

Background

This Chapter summarises the background knowledge needed for this dissertation, introducing fundamental concepts of probability theory [91], machine learning [42], neural networks [42] and natural language processing [45].

2.1 Probability

In this section I briefly review probability concepts that underpin the methods used throughout this document—basic probability and Bayes' theorem, the foundation of the Naïve Bayes classified used in Chapters 4 and 5; and the probability distributions mentioned in further chapters.

2.1.1 Basic probability concepts

Experiments, outcomes and events The sample space Ω defines the set of all outcomes $\omega \in \Omega$ of an experiment. One example of experiment is the tossing of two coins. In this case, the sample space and the possible outcomes would be

$$\Omega = \{\omega_1 = (\text{heads, tails}), \omega_2 = (\text{heads, heads}), \omega_3 = (\text{tails, heads}), \omega_4 = (\text{tails, tails})\}\,.$$

The subsets of Ω are called events, denoted with uppercase letters. In the previous example, one event would be "both coins land on the same side". If we call it A, we would have the following notation:

$$A = \{\omega_2, \omega_4\} \subset \Omega\,.$$

Random variable and probability distribution Random variables are functions that map the outcomes of a sample space to real numbers. Formally, a random variable X

is a function $X : \Omega \to \mathbb{R}$. The probability that X takes on the value x is

$$P(X = x) = P(\{\omega \in \Omega | X(\omega) = x\}) . \tag{2.1}$$

If it is obvious which random variable is being referenced, $P(X = x)$ is shortened to $P(x)$. The probability distribution of a random variable describes the probabilities of each of its values.

Joint probability distribution and conditional probability A joint probability distribution defines the probabilities of multiple random variables assuming simultaneous values. For example, given random variables X and Y, $P(X = x, Y = y)$ is the probability of events $\{\omega \in \Omega_X | X(\omega) = x\}$ and $\{\omega \in \Omega_Y | Y(\omega) = y\}$ happening. When the random variables in question are unambiguous, $P(x, y)$ is the favoured notation.

It is often desirable to predict probabilities of events occurring given that some other event happened (e.g. the probability of raining in the evening given that it is sunny in the morning or the probability that a masked woman has broken into my house given that my door lock is broken). The probability of event A happening given that event B occurs is known as the conditional probability of A given B and is defined as

$$P(A|B) = \frac{P(A, B)}{P(B)} . \tag{2.2}$$

Independence The events A and B are independent if the occurrence of one does not affect the probability of the other:

$$P(A|B) = P(A) .$$

Or, equivalently,

$$P(B|A) = P(B) .$$

In addition, two random variables X and Y are independent if, and only if, their joint probability distribution is the product of their individual distributions. That is:

$$P(x, y) = P(x)P(y), \ \forall x, y . \tag{2.3}$$

Bayes' Theorem The Bayes's Theorem is the basis of one of the machine learning classifiers we use in this dissertation—the aptly named Naïve Bayes. This theorem is useful when we can estimate from data $P(x|y)$ (e.g. the probability of the word "cat" appearing in a document about the cold war), $P(y)$ (the probability that a document is about the

cold war) and $P(x)$ (the probability that the word "cat" appears in a document), and wish to infer $P(y|x)$ (the probability of a document containing the word "cat" being about the cold war). In this case, Bayes' theorem states that

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}\,. \tag{2.4}$$

$P(y|x)$, $P(x|y)$ and $P(y)$ are known, respectively, as the posterior, the likelihood and the prior probabilities.

Discrete and continuous distributions A random variable X has a discrete probability distribution when it assumes a countable[1] number of values (e.g. rolls of a dice). In this case, the probability mass function (PMF) gives the probabilities for each value of X. If a random variable Y can assume any value in a continuous range (e.g. lifespan of a lamp), the probability density function (PDF) $f_Y(y)$ can be used to calculate the probability of Y assuming a value in the range $[a, b]$:

$$P(y \in [a, b]) = \int_a^b f_Y(y)dy\,. \tag{2.5}$$

We will denote the PMF of X as $p_X(x)$ and the PDF of Y as $f_Y(y)$; we drop the subscript when doing so does not result in ambiguity.

Expectation The expectation of a random variable X, $\mathbb{E}[X]$, is the mean of the values it can assume weighted by their respective probabilities. The expectation is defined, in the discrete and continuous cases respectively, as:

$$\mathbb{E}[X] = \sum_x xp(x) \tag{2.6}$$

$$\mathbb{E}[X] = \int_{\mathbb{R}} xf(x)dx\,. \tag{2.7}$$

The expectation of X can also be referred as the mean of X, μ_X, or simply μ if the random variable is unambiguous.

Variance The variance of a random variable X, $\mathrm{Var}(X)$, measures how much its values are spread out from the mean. The variance is defined as:

$$\mathrm{Var}(X) = E[(X - \mu)^2]\,. \tag{2.8}$$

[1] That is, has the same cardinality of some subset of $\mathbb{N}$.

A convenient formula to compute the variance can be inferred from the definition:

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \,. \tag{2.9}$$

The square root of the variance of X is called the standard deviation of X and is denoted as σ_X (or just σ if adequate).

Covariance The covariance of two random variables measures their joint variability and is defined as:

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \,. \tag{2.10}$$

This can be simplified to the formula below:

$$\mathrm{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \,. \tag{2.11}$$

The covariance of a random variable with itself is its variance:

$$\mathrm{Cov}(X, X) = \mathrm{Var}(X) \,. \tag{2.12}$$

2.1.2 Relevant distributions

Here we briefly describe the probability distributions mentioned throughout this dissertation. The notation used to state that X follows a distribution P with parameter θ is $X \sim P(\theta)$.

Bernoulli The Bernoulli distribution models experiments that can assume only two values (e.g. failure and success), such as the number of heads in one flip of a coin. It is governed by the parameter p, $0 \leq p \leq 1$, the probability of "success". Let $X \sim \mathrm{Bernoulli}(p)$, then:

$$p(x) = \begin{cases} p, & x = 1 \\ 1 - p, & x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

$$\mu = p \tag{2.14}$$

$$\sigma^2 = p(1 - p) \,. \tag{2.15}$$

Binomial The binomial distribution is the sum of n independent Bernoullis, such as the number of heads in 100 coin flips. It is governed by the parameters p, success chance, and n the number of trials. Let $X \sim \mathrm{Bin}(n, p)$, then:

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x} \tag{2.16}$$

$$\mu = np \tag{2.17}$$

$$\sigma^2 = np(1-p)\,. \tag{2.18}$$

Multinomial The multinomial distribution generalises the binomial distribution to trials with more than two outcomes, such as a series of die rolls. Its parameters are $p_1, \ldots, p_k$, the outcomes' probabilities, and n, the number of trials. Let $x_1, \ldots, x_k$ be the number of occurrences of each outcome, $X_1, \ldots, X_k$ the random variables for each outcome, $\mathbf{X} = [X_1 \ldots X_k]^\mathbf{T}$ and $\mathbf{X} \sim \text{Multinomial}(n, p_1, \ldots, p_k)$, then:

$$p(x_1, \ldots, x_k) = \frac{n!}{x_1! \ldots x_k!} p_1^{x_1} \ldots p_k^{x_k} \tag{2.19}$$

$$\boldsymbol{\mu} = n \begin{bmatrix} p_1 \\ \vdots \\ p_k \end{bmatrix} \tag{2.20}$$

$$\boldsymbol{\Sigma} = n \begin{bmatrix} p_1(1-p_1) & -p_1 p_2 & \cdots & -p_1 p_k \\ -p_2 p_1 & p_2(1-p_2) & \cdots & -p_2 p_k \\ \vdots & \vdots & \ddots & \vdots \\ -p_k p_1 & -p_k p_2 & \cdots & p_k(1-p_k) \end{bmatrix} \tag{2.21}$$

Note that $\mathbf{X}$ is a vector of random variables, so it has a mean vector $\boldsymbol{\mu}$ and a covariance matrix[2] $\boldsymbol{\Sigma}$ instead of scalar values.

Poisson The Poisson distribution models the number of events occurring in a fixed interval of time or space, given that those events occur with a constant mean rate λ and are independent of the time since the last event; for example, the number of patients arriving in a hospital in a day. Let $X \sim \text{Poisson}(\lambda)$, then:

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{2.22}$$

$$\mu = \lambda \tag{2.23}$$

$$\sigma^2 = \lambda\,. \tag{2.24}$$

[2]If we represent the covariance matrix cell with coordinate (i, j) as $\sigma^2_{i,j}$, than $\sigma^2_{i,j} = \text{Cov}(X_i, X_j)$. Thus, if $i = j$, $\sigma^2_{i,j} = \text{Var}(X_i)$.

Uniform The uniform distribution models random variables that are equally likely to assume any value in a continuous interval $[a, b]$. Let $X \sim \mathrm{U}(a, b)$, then:

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

$$\mu = \frac{a+b}{2} \tag{2.26}$$

$$\sigma^2 = \frac{(b-a)^2}{12}\,. \tag{2.27}$$

Beta The beta distribution is a continuous distribution that models random variables that assume values in the interval $[0, 1]$. It is characterized by two shape parameters α and β. Its range of values enables it to be a good model for proportions and probabilities. Let $X \sim \mathrm{Beta}(\alpha, \beta)$, then:

$$f(x) = \begin{cases} \dfrac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, & 0 < x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.28}$$

$$\mu = \frac{\alpha}{\alpha+\beta} \tag{2.29}$$

$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}\,, \tag{2.30}$$

where Γ is the Gamma Function, which generalises the factorial to all complex numbers, except for non-positive integers. It is defined as:

$$\Gamma(z) = \begin{cases} (z-1)!, & z \in \mathbb{Z}^+ \\ \int_0^\infty x^{z-1}e^{-x}dx, & \text{z has a positive real part} \end{cases} \tag{2.31}$$

Dirichlet The Dirichlet distribution generalises the Beta distribution to multiple variables: it generates sample vectors of k entries such that the value of all entries add up to one—which makes it useful to model categorical probability distributions. It is parameterized by the vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_k]^{\mathbf{T}}$ of positive reals. Let $\mathbf{X}$ be a random vector

of k components, $\alpha_0 = \sum_{i=1}^{k} \alpha_i$, and $\mathbf{X} \sim \text{Dir}(\boldsymbol{\alpha})$, then:

$$f(x_1, \ldots, x_k) = \begin{cases} \dfrac{\Gamma(\alpha_0)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}, & \sum_{i=1}^{k} x_i = 1 \text{ and } x_i \geq 0 \; \forall i \in 1, \ldots, k \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

$$\boldsymbol{\mu} = \frac{1}{\alpha_0} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} \tag{2.33}$$

$$\boldsymbol{\Sigma} = \frac{1}{a_0^2(\alpha_0 + 1)} \begin{bmatrix} \alpha_1(\alpha_0 - \alpha_1) & -\alpha_1\alpha_2 & \cdots & -\alpha_1\alpha_k \\ -\alpha_2\alpha_1 & \alpha_2(\alpha_0 - \alpha_2) & \cdots & -\alpha_2\alpha_k \\ \vdots & \vdots & \ddots & \vdots \\ -\alpha_k\alpha_1 & -\alpha_k\alpha_2 & \cdots & \alpha_k(\alpha_0 - \alpha_k) \end{bmatrix} \tag{2.34}$$

2.2 Machine learning

We now introduce the basic machine learning (ML) concepts and models we mention in subsequent chapters.

2.2.1 Basic machine learning concepts

Feature representation We call each unit of input to a ML model an example (e.g. a text or an image). Each example is represented as a vector $\mathbf{x} \in \mathbb{R}^d$ of d features. These features can be pixel values in an image or word frequencies in natural language documents, for example. A collection of n examples is denoted as a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where each row contains one of the examples.

Supervised and unsupervised We consider two main learning categories: supervised and unsupervised. In the first case, for each example $\mathbf{x}^{(i)}$, $i = (1, \ldots, n)$, there is a label $y^{(i)}$ that indicates to which *class* or *category* the example belongs (for image classification, examples of labels would be "cat" and "dog"). A collection of n labels can be denoted as a vector $\mathbf{y} \in \mathbb{R}^n$, where the component $\mathbf{y}_i$ is the label for $\mathbf{x}^{(i)}$. In the unsupervised learning case, there are no labels; instead, the model learns properties of the data, such as the probability distribution that generated it, or performs tasks such as clustering, where the objective is to divide the data into groupings of similar examples.

Classification and regression Supervised learning tasks can be further categorised into two types: classification and regression. In the former case, the labels belong to a set of predefined cases (is this movie review positive, negative or neutral?); in the latter, the

labels belong to a continuous range of values (what rating do we predict for this movie review?).

2.2.2 Relevant classifiers

We now present the shallow (not neural network based) classifiers we will use throughout this dissertation. We employ Naïve Bayes, Support Vector Machine and Gradient Boosted Trees for document classification in Chapter 4, and linear-chain Conditional Random Fields for sequence labelling tasks in Chapters 3 and 4. We also briefly discuss decision tree ensembles as we consider it background knowledge for Gradient Boosted Trees.

Naïve Bayes Naïve Bayes (NB) are ML classifiers that assume independence between features. NB computes through Bayes' Theorem the conditional probability that a example belongs to a class given its features. That is, given an example $\mathbf{x} = [x_1, \ldots, x_n]$ and a set of categories C:

$$P(c|\mathbf{x}) = \frac{P(\mathbf{x}|c)P(c)}{P(\mathbf{x})},\ \forall c \in C\,, \tag{2.35}$$

as Bayes' theorem states. In this dissertation we use NB classifiers to classify text documents, in which case documents of a given class are assumed to have been generated by a multinomial distribution. That is, given a vocabulary V of $\mathcal{V}$ words, a document $\mathbf{x} \in \mathbb{R}^{\mathcal{V}}$, where the i-th dimension represents the number of times the i-th word in the vocabulary appears in the document, and $p_{c1}, \ldots, p_{c\mathcal{V}}$, the probabilities of each word appearing in a document of class c, the likelihood is [115]:

$$P(\mathbf{x}|c) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ci}^{x_i}\,. \tag{2.36}$$

Then, the probabilities p_{ci} can be estimated by the frequencies observed in the training data, while the priors $P(c)$ can be assumed to be equiprobable or also estimated during training.

This method has a generalisation problem: if a word w never appears in a document of a given class c during training, at test time a document that contains w will have zero probability of belonging to class c, since the conditional probability is proportional to $p_{cw} = 0$. Such behaviour is undesirable, as the classifier would not even consider any of the other probabilities. To resolve that, we can add a positive number α to all word occurrences, so that no probability will be equal to zero. This approach is called Laplace smoothing.

Linear support vector machine Linear Support Vector Machine is a particular case of SVM [12, 25] that uses a linear kernel[3], i.e. the dot product operation. Given a set of n linearly separable training examples and corresponding labels $(\mathbf{x}^{(1)}, y^{(1)}) \ldots, (\mathbf{x}^{(n)}, y^{(n)})$, each $y^{(i)} \in \{-1, 1\}$, the aim is to find the two most distant parallel hyperplanes that separate positive and negative classes. The region between them is called the margin, and the hyperplane that lies halfway between them is the classifier's decision boundary. The following equations describe the hyperplanes:

$$\mathbf{w}^T\mathbf{x} - b = 1 \text{ , such that positive classes are on or above this boundary;} \tag{2.37}$$

$$\mathbf{w}^T\mathbf{x} - b = -1 \text{ , such that negative classes are on or below this boundary;} \tag{2.38}$$

$$\mathbf{w}^T\mathbf{x} - b = 0 \text{ , the decision boundary.} \tag{2.39}$$

Note that the distance between the boundaries is $\frac{2}{\|\mathbf{w}\|}$.

Alas, it is not always the case that the data is linearly separable—in this case, besides maximizing the margin, the aim is to minimise the number of examples that lie in the margin or are misclassified. To penalise such cases, the hinge loss can be used, which is defined as $l = \max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}))$. When the example lies to the appropriate side of the margin, the loss is zero; otherwise, it is proportional to the distance of the sample to its correct hyperplane boundary.

Therefore, training a Linear SVM model means finding $\mathbf{w}$ and b that minimises the following expression:

$$\frac{1}{2}\|\mathbf{w}\|^2 + c\frac{1}{n}\left[\sum_i \max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}))\right], \tag{2.40}$$

where c is a regularisation parameter that controls the trade-off between the two objectives: maximising the margin and penalising incorrect predictions.

In this dissertation, for tasks with more than two categories we use a one-vs-rest approach, where we train one classifier for each class. In other words, instead of one task with k classes, we have k task with 2 classes: the positive class is one of the original categories and the negative class is the set of all other examples.

Decision tree ensembles Decision trees are widely used models due to their simplicity and interpretability. Each internal node in a decision tree is a predicate about the data that subdivides it into two subtrees; for example, one node may be "is male", in which case it will be the root of two subtrees—one for male examples and another for female examples. The leaves of the tree give the possible classification outcomes. Learning consists in finding

[3]Please refer to Goodfellow et al. [42, p. 139] for an explanation about kernels.

the nodes that better split the data. The interpretability of decision trees stem from the fact that, for each prediction, we can immediately obtain the series of decisions that led to it.

Instead of using only one tree, we can train several trees by using different subsets of our data and, at test time, average the outputs of all trees to compute the final prediction. In addition to sampling subsets of the data, we can create different models by sampling subsets of features. When both methods are used, the ensemble is called a random forest [15].

Gradient boosted trees Averaging the prediction of several weaker models to obtain a stronger model is not the only way to create a ensemble. One can use a boosting [125] approach:

1. Train a model;
2. Fit another model on the residuals[4] of the previous model;
3. Repeat step 2 using the residuals of the last model trained until some stopping criterion is reached.

To combine the predictions, we simply compute the sum of the outputs of each weak model. When the weak models are decision trees, the ensemble is called gradient boosted trees, which is the method implemented in the XGBoost [22] library we use in Chapter 4.

Linear-chain conditional random fields Conditional random fields [75] are a class of probabilistic models commonly used for sequence labelling. Let $\mathbf{X}$ be a random vector over data sequences (natural language sentences for example) and $\mathbf{Y}$ a random vector over corresponding sequences of labels (part-of-speech tags or named entity labels, for example), where all components $\mathbf{Y}_i$ of $\mathbf{Y}$ belong to a finite set of classes. Then, the CRF framework constructs a conditional model $P(\mathbf{Y}|\mathbf{X})$ from training sentences, defined [75] as follows:

> Let $G = (V, E)$ be a graph such that $\mathbf{Y} = (\mathbf{Y}_v)a_{v \in V}$, so that $\mathbf{Y}$ is indexed by the vertices of G. Then $(\mathbf{X}, \mathbf{Y})$ is a **conditional random field** in case, when conditioned on $\mathbf{X}$, the random variables $\mathbf{Y}_v$ obey the Markov property with respect to the graph: $P(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G.

Linear-chain CRFs are a special case where G is a chain—$G = (V = \{1, \ldots, m\}, E = \{(i, i + 1)\})$—that is, a label is conditioned only on the input and on its immediate

[4]The difference between ground truth and predictions values.

neighbour labels. In this case, we can state the following about the distribution over the label sequence $\mathbf{Y}$ given input sequence $\mathbf{X}$:

$$P(\mathbf{y}|\mathbf{x}) \propto \exp\left(\sum_{e\in E,k} \lambda_k f_k(e, \mathbf{y}|_e, \mathbf{x}) + \sum_{v\in V,k} \mu_k g_k(v, \mathbf{y}|_v, \mathbf{x})\right), \tag{2.41}$$

where $\mathbf{x}$ is a data sequence, $\mathbf{y}$ is a label sequence, $\mathbf{y}|_S$ is the set of components of $\mathbf{y}$ associated with the vertices in subgraph S, and f_k and g_k are given feature functions.

Training linear-chain CRF means finding the optimal parameters $\theta = (\lambda_1, \ldots; \mu_1, \ldots)$[5]. Then, given a input sequence $\mathbf{x}$, we predict the label sequence $\hat{\mathbf{y}}$ that maximises $P(\mathbf{y}|\mathbf{x})$:

$$\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})\,. \tag{2.42}$$

2.3 Neural networks

Neural networks are compositions of parametrized functions (or layers) whose objective is to approximate some other function. For example, let $f^*(\mathbf{x})$ be the function that maps all possible images of cats or dogs to one of those categories. Then, the goal of a neural network classifier $f(\mathbf{x}; \boldsymbol{\theta})$ is to learn the parameter vector $\boldsymbol{\theta}$ that best approximates the function f^*.

2.3.1 Layers and activation functions

The last layer of a neural network is the *output layer*. The others are the *hidden layers*. We usually call the output of hidden layers *activations*. Table 2.1 exemplifies terminology and notation for a neural network with one hidden layer.

Table 2.1: Notation for a neural network with one hidden layer.

Notation	Meaning
$\mathbf{x}$	Input
f_1	Hidden layer
$\mathbf{a} = f_1(\mathbf{x})$	Hidden layer activation
f_2	Output layer
$f_2(\mathbf{a}) = f_2(f_1(\mathbf{x}))$	Neural network output

Each layer is a matrix multiplication between its input and its parameters followed by a nonlinear transformation, commonly referred to as an activation function. Since the output components are a result of an operation over all input components, these layers are called fully connected layers. They are parametrized by a weight matrix $\mathbf{W} \in \mathbb{R}^{d\times n}$

[5]Refer to Lafferty et al. [75] for details on parameter estimation.

and a bias vector $\mathbf{b} \in \mathbb{R}^d$, where n and d are the dimensionality of the layer input and output[6], respectively. A fully connected layer computes the following equation:

$$\mathbf{a} = g(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{2.43}$$

where $g(\cdot)$ is an activation function. Since compositions of linear functions are also linear functions, activations functions are responsible for the expressive power of neural networks. In this work we mention four activation functions: sigmoid, hyperbolic tangent, rectified linear unit (ReLU) and softmax.

Sigmoid

The sigmoid function "squashes" a real-valued input into the $(0, 1)$ interval (Figure 2.1):

$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}. \tag{2.44}$$

Therefore, one can use the sigmoid function to produce probabilities. So, it is often used as the activation function of the output layer for binary classification tasks. When used as the activation function of a hidden layer, this function can become troublesome, since its derivative rapidly approaches zero as the input absolute value increases, which is a major obstacle for gradient-based learning.

Hyperbolic tangent

The hyperbolic tangent function is similar to the sigmoid function, but squashes the input into the $(-1, 1)$ interval (Figure 2.2):

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{2.45}$$

The derivative of the hyperbolic tangent also approaches zero as the input absolute value increases. In the context of this work, the hyperbolic tangent is used as the activation function of one of the LSTM gates (Section 2.3.3).

Rectified linear unit

The rectifier linear unit (ReLU) [99] (Figure 2.3) is defined as:

$$\text{ReLU}(z) = \max(0, z). \tag{2.46}$$

[6]We refer to the output dimensionality as the layer's dimensionality, size or number of units.

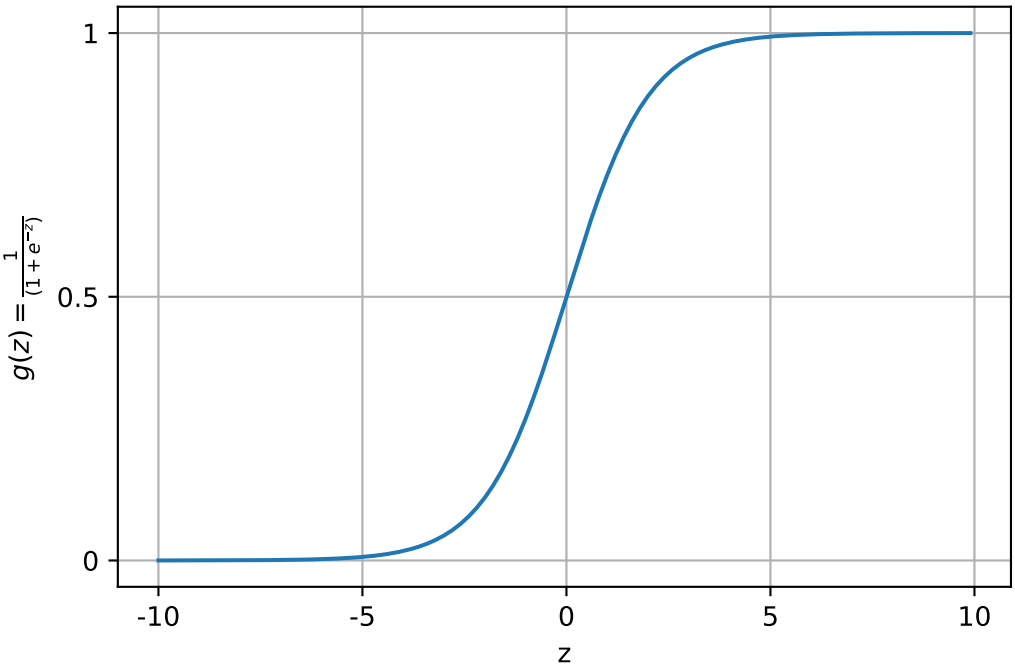


Figure 2.1: The sigmoid activation function.

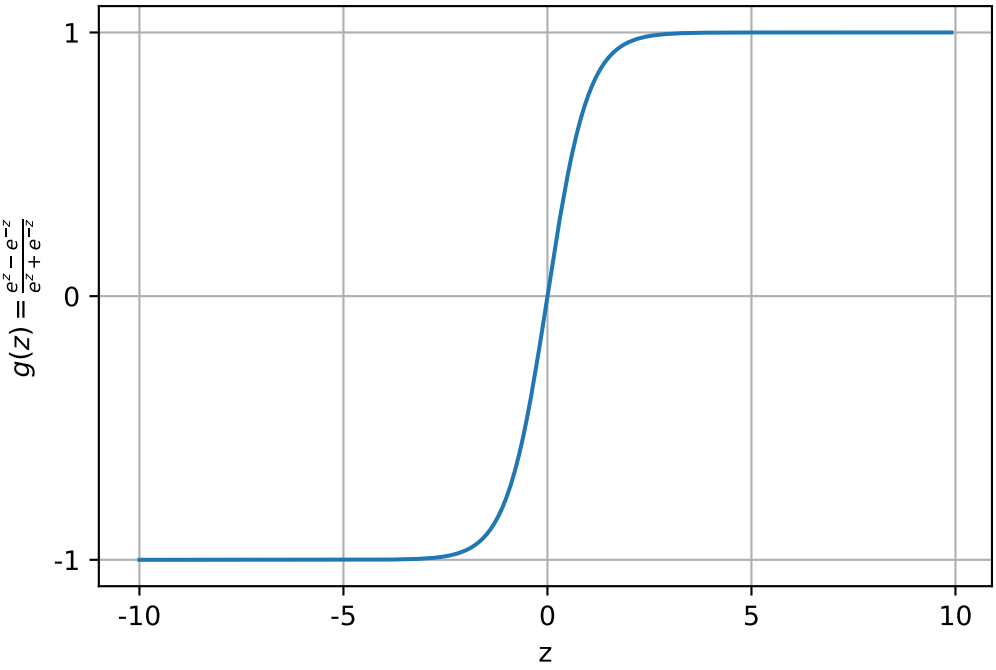


Figure 2.2: The tanh activation function.

It was widely adopted by computer vision and natural language communities as the activation function of hidden layers, since it empirically works well. This is in part due to its easily computable derivative[7] and to it not saturating when processing large positive numbers.

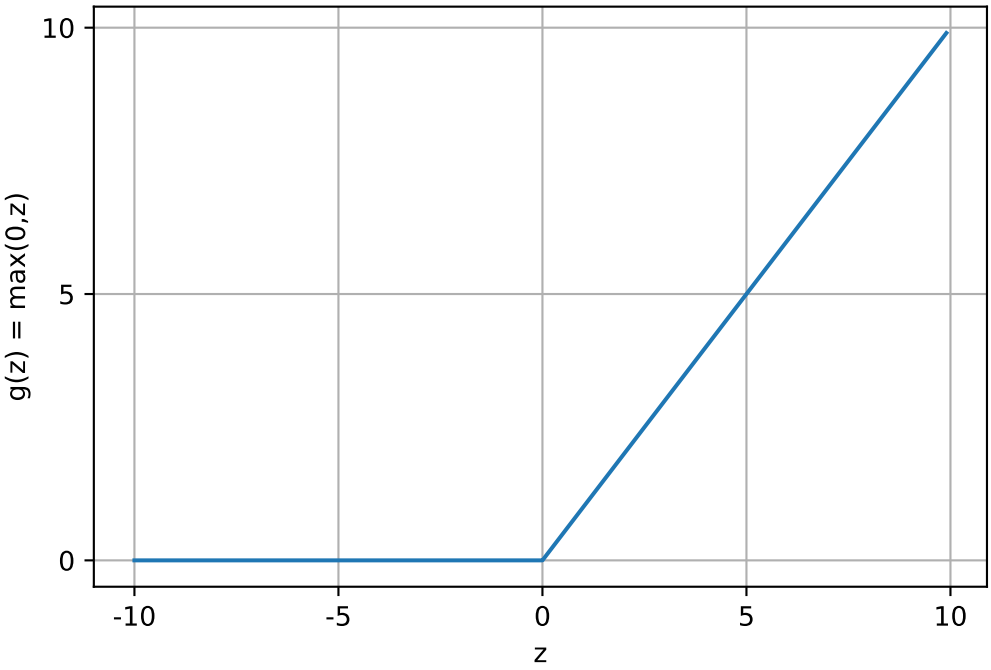


Figure 2.3: The ReLU activation function.

Softmax

The softmax function is defined as:

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \text{ for } i = 1, \ldots, k \text{ and } \mathbf{z} = (z_1, \ldots, z_k)\,. \tag{2.47}$$

That is, the exponential function is applied to each component of the input vectors and then these values are normalised. Since each component of the output vector is in the interval $(0, 1)$ and they add up to 1, the softmax output can be interpreted as a probability distribution. For that reason, softmax is widely used as the activation function of the output layer for multi-class classification tasks.

[7]The ReLU function is non-differentiable at zero. In practice, this is not a problem since one can use the right (1) or left (0) derivatives [119, p. 36].

2.3.2 Parameter learning

The neural network parameters are learned through minimisation of a cost function $J(\boldsymbol{\theta})$, a process we call model training. For classification tasks, the negative log likelihood loss is generally used. Let $\mathbf{z}$ be a vector of predicted probabilities (e.g. the softmax output) for input $\mathbf{x}$, and y the target class. Then the negative log likelihood loss function is:

$$\text{nll}(y; \mathbf{z}|\boldsymbol{\theta}, \mathbf{x}) = -\log \mathbf{z}_y \,. \tag{2.48}$$

Figure 2.4 illustrates how the negative likelihood loss function pensalises classification errors. Essentially, the loss is 0 when the correct class is predicted with absolute certainty and increases as the predicted probability decreases, approaching infinity as the predicted probability approaches zero.

The cost function to be minimised is the average negative log likelihood loss over the training examples:

$$J(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \text{nll}(y^{(i)}; \mathbf{z}^{(i)}|\boldsymbol{\theta}, \mathbf{x}^{(i)}) \,. \tag{2.49}$$

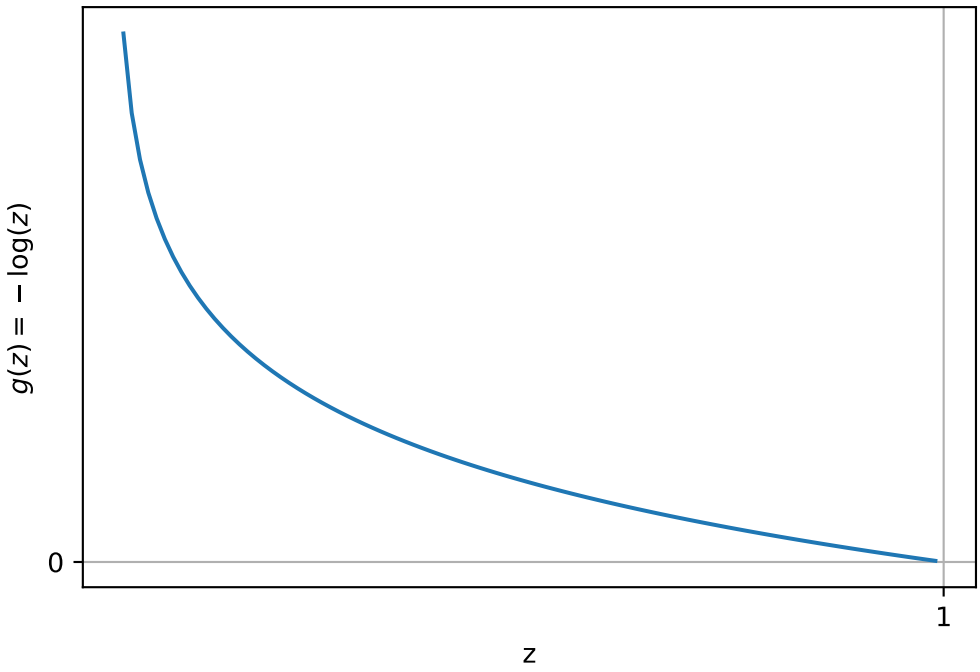


Figure 2.4: The negative log likelihood loss function.

The gradient descent algorithm is used to minimise the cost function. It updates the parameters $\boldsymbol{\theta}$ in the opposite direction of the gradient of the cost function with respect to

$\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \cdot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})_t \,, \tag{2.50}$$

where η, the learning rate, is a hyperparameter that controls the size of the descent step.

Computing the gradient of the cost function is expensive, since it requires getting predictions for all training examples. In practice, the gradient is approximated using a mini-batch of m examples, an approach called stochastic gradient descent (SGD) with mini-batches.

The backpropagation algorithm [120] computes the gradient. It is an efficient application of the chain rule of calculus that starts from the last layer and goes backward through the network, computing the gradient a layer at a time and avoiding superfluous operations.

Throughout this dissertation we use variants of SGD, constructed to find better local minima and converge faster: SGD with momentum and Adam.

SGD with momentum

Momentum [120, 107] accelerates SGD convergence by using a descent step that is a linear combination of the current and previous learning steps, which dampens oscillations that slow convergence [117]:

$$\Delta\boldsymbol{\theta}_t = \beta \cdot \Delta\boldsymbol{\theta}_{t-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})_t \tag{2.51}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \Delta\boldsymbol{\theta}_t \,. \tag{2.52}$$

The hyperparameter β, $\beta \geq 0$, controls the relative contribution of past and current gradients.

Adam

Adam [69], in addition to momentum, uses adaptive learning rates—different values for different parameters. It does so by keeping a running average of past gradients $\mathbf{m}_t$ (estimate of the first moment of the gradient) and past squared gradients $\mathbf{v}_t$ (estimate of the second

moment of the gradient):

$$\mathbf{m}_t = \beta_1 \cdot \mathbf{m}_{t-1} + (1 - \beta_1) \cdot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})_t \tag{2.53}$$

$$\mathbf{v}_t = \beta_2 \cdot \mathbf{v}_{t-1} + (1 - \beta_2) \cdot \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})_t^2 \tag{2.54}$$

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t} \tag{2.55}$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t} \tag{2.56}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon} \hat{\mathbf{m}}_t \,, \tag{2.57}$$

where $\hat{\mathbf{m}}_t$ and $\hat{\mathbf{v}}_t$ correct the bias introduced by initialising $\mathbf{m}_t$ and $\mathbf{v}_t$ with vectors of zeroes, ϵ is a small scalar that prevents division by 0, and β_1 and β_2, $0 \leq \beta_1, \beta_2 < 1$, are factors that control the contribution of past gradients. All operations involving vectors are done element-wise.

Learning rate tuning

Choosing a good learning rate is paramount for learning—too high a value and gradient descent may diverge; too low, and training may take too long. In this dissertation we often employ the learning rate range test [131] to find an adequate value. The method consists in training the model for a few iterations, starting from a low learning rate value and exponentially increasing it at each iteration. The obtained losses are then plotted against the corresponding learning rates, as exemplified by Figure 2.5. A good learning rate would be close to the point where the loss starts to increase: high enough for faster learning, but not so high as to impede it.

Fine-tuning

Instead of initialising a neural network with random parameters, one can start from a pretrained representation—a previously trained model—and further train it on the task of interest, an approach called fine-tuning. For example:

1. A model with randomly initialised parameters is trained on texts in Portuguese from many different sources (Wikipedia, web pages, books) to approximate the function $f_{\text{port}}(s)$, which gives the probability of a sentence s occurring in the Portuguese language (source task).

2. A model initialised with the parameters learned in step 1 is trained—fine-tuned—on texts from legal documents to approximate the function $f_{\text{legal\_port}}(s)$, which gives the probability of a sentence s occurring in legal documents in the Portuguese language (target task).

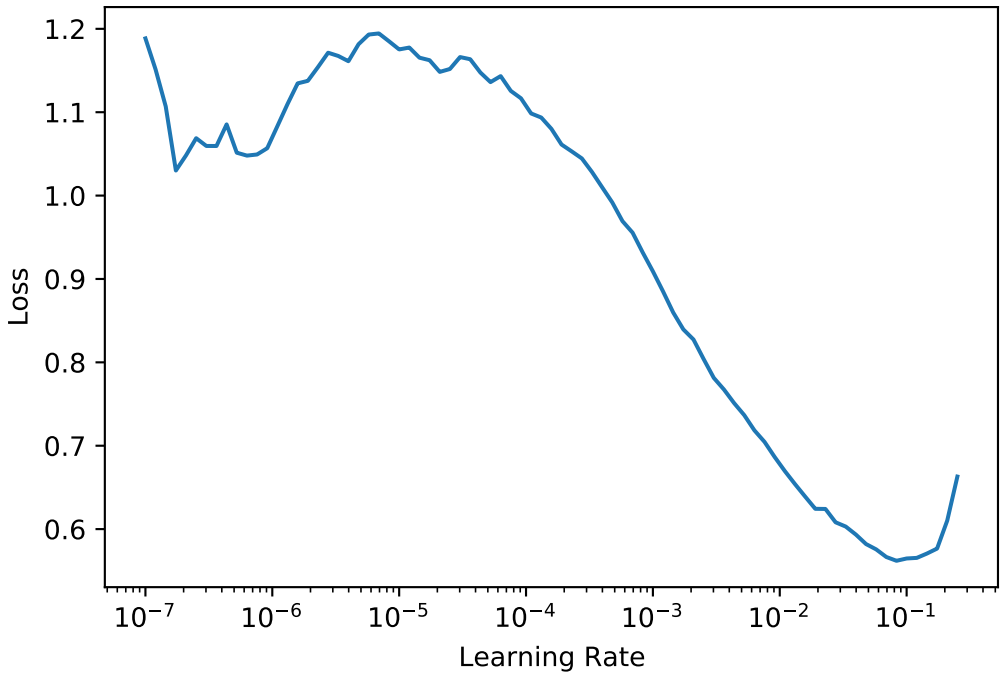


Figure 2.5: An example of a loss against learning rate plot for a range test. A good choice for learning rate in this case would be around 10^{-2}.

Through model fine-tuning it is possible to achieve more accurate models with fewer gradient descent steps, particularly when source and target tasks are similar [118].

In this dissertation, fine-tuning is used to adapt word embeddings and language models (§2.4.4) from general to target domain (§3.3, §5.3.3), and to transfer a model trained on an object recognition task to a document classification task (§4.3).

2.3.3 Special networks

Some neural networks are composed by layers other than fully connected ones. In the context of this work, we use two of these special networks: recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

Recurrent neural networks

Recurrent neural networks are used to process sequential data, such as natural language texts. While other types of neural networks can process sequential data of fixed length, RNNs can do so with variable-length sequences using its internal state. A basic RNN layer [33] can be described as a regular fully connected layer that at each time step t processes the input $\mathbf{x}$ and the hidden state $\mathbf{h}$ of the previous time step:

$$\mathbf{h}_t = g_h(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2.58}$$

$$\mathbf{a}_t = g_a(\mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a)\,, \tag{2.59}$$

Where $\mathbf{W}_h$, $\mathbf{U}_h$ and $\mathbf{W}_a$ are learnable weights, $\mathbf{b}_h$ and $\mathbf{b}_a$ are learnable biases, $g_h(\cdot)$ and $g_a(\cdot)$ are activation functions and $\mathbf{a}_t$ is the layer output at each time step.

Long short-term memory (LSTM) networks [50] use a recurrent layer with additional mechanisms that mitigate the vanishing gradient problem[8], enabling information to be retained for longer. These mechanisms are called "gates". There are four of them: the input ($\mathbf{i}_t$), cell ($\mathbf{g}_t$), output ($\mathbf{o}_t$), and the later introduced [40] forget ($\mathbf{f}_t$) gates. The LSTM layer computes the following equations:

$$\mathbf{i}_t = \text{sigmoid}(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_{hi}) \tag{2.60}$$

$$\mathbf{f}_t = \text{sigmoid}(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_{hf}) \tag{2.61}$$

$$\mathbf{g}_t = \tanh(\mathbf{W}_{ig}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_{hg}) \tag{2.62}$$

$$\mathbf{o}_t = \text{sigmoid}(\mathbf{W}_{io}\mathbf{x}_t + \mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_{ho}) \tag{2.63}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \tag{2.64}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)\,, \tag{2.65}$$

Where $\odot$ is the element-wise multiplication and $\mathbf{c}_t$ is the memory cell state, the mechanism responsible for retaining (long) short-term memory. Perhaps a little confusingly, the hidden state $\mathbf{h}_t$ serves as the output. Figure 2.6 illustrates the LSTM layer.

Convolutional neural networks

Convolutional neural networks (CNN) [79] are mainly used to process data with a grid-like topology [42]—where there is a spatial correlation between neighbourhood regions of the input, such as pixels in a image and words in a text. Instead of the regular matrix multiplication of fully connected layers, the convolutional layer process its input through a number of filters that learn to identify features such as edges and shapes in case of images, and informative combinations of words in case of texts. Depth is essential for CNNs: higher layers learn to identify features as complex as the presence of faces and certain objects. The use of CNN is pervasive in computer vision applications and was

[8] When the gradient of the cost function gets too small and the parameters cannot be updated through gradient descent.

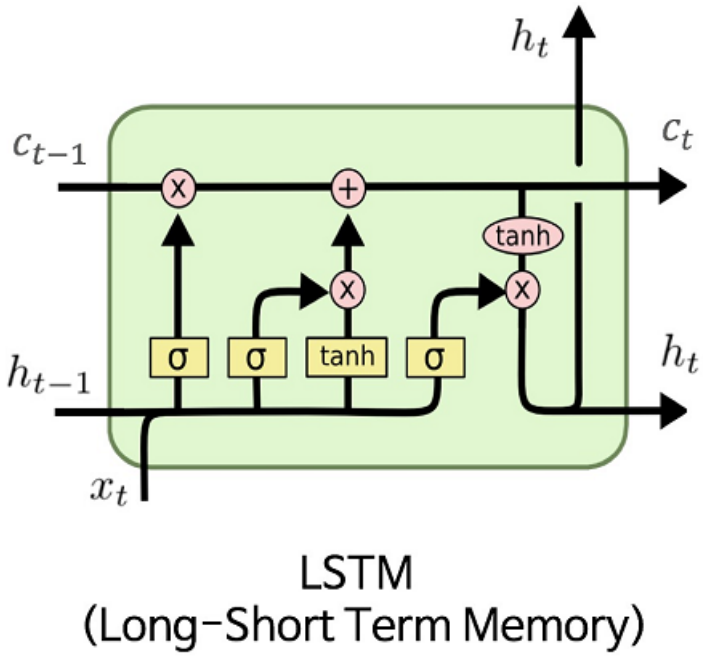


Figure 2.6: The LSTM layer. The symbol σ denotes the sigmoid function. Image by MingxianLin CC-BY-SA-4.0. Available at `https://commons.wikimedia.org/wiki/File:LSTM.png`.

responsible for many recent advances in the state of the art of various tasks (e.g. image classification, instance segmentation, image captioning).

Pooling operations

Pooling operations are used to reduce the dimension of feature maps and making representation invariant to small translations. They work by dividing a feature map into sub-regions and independently down-sampling the features in each of them to a single value: the average (average pooling) and the maximum (max pooling) are two examples.

Global pooling reduces all feature maps to 1 value each, again, using the maximum or average value for example. In this work we apply global pooling to the output of the last convolutional layer to obtain feature vectors for downstream tasks, concatenating the vectors resulting from both global max pooling and global average pooling.

2.4 Natural language processing

Natural language processing (NLP) is the field of study concerned with enabling computers to process natural language data. It encompasses tasks such as text classification, named entity recognition, speech recognition and text summarisation. The first step in a text processing workflow is tokenising the text (§2.4.1). Then, the tokens are aggregated in

some form to construct a representation of the text. One example of that are bag-of-words models (§2.4.2). Once the representation method is chosen and a model is trained on the data, the performance can be evaluated by measuring metrics (§2.4.3). Model performance can be improved by leveraging natural texts in a unsupervised way using transfer learning techniques (§2.4.4).

2.4.1 Tokenisation

The raw text input first needs to be subdivided into smaller parts (tokens), a process called tokenisation. The smaller units can be words, sub-words or characters. Choosing between words and character units is a trade-off between the natural semantics of the former and the smaller vocabulary of the latter. Since the set of characters is much smaller than the set of words, choosing characters also reduces the chance of finding out-of-vocabulary tokens at test time. Subword tokenisation [73, 148] promises to deliver the best of both worlds by discovering the sequences of characters that most frequently occur in the training texts, resulting in a smaller vocabulary of often semantically interpretable units.

2.4.2 Bag-of-words model

A tokenised text can be represented by a vector $\mathbf{x} \in \mathbb{R}^{\mathcal{V}}$, $\mathcal{V}$ the vocabulary size, where the i-th component, $i \in (1, \ldots, \mathcal{V})$, is the number of times the i-th token appears in the text. This is called a bag-of-words model: the text is represented by the tokens that appear in it, with no regard for word order. In addition to word counts, bag-of-words models can use as a feature the term frequency-inverse document frequency (tf-idf) value, which weighs token importance according to how often it appears in a given text and how seldom it appears in the corpus (collection of texts). We compute it as follows:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t) \tag{2.66}$$

$$\text{idf}(t) = \log \frac{1 + n}{1 + \text{count}(t)} + 1\,, \tag{2.67}$$

where $\text{tf}(t, d)$ is the frequency of token t in document (text) d, n is the total of documents in the corpus, and $\text{count}(t)$ is the number of documents that contain term t.

One can also use sequences of tokens as vocabulary entries, such as `(drink, milk)` and `(not, very, funny)`. These are called n-grams, where n is the number of words in the sequence. N-grams enable richer representations at the cost of an exponentially growing vocabulary (and, consequently, increased feature dimensionality). An example of

richer representation is the ability to capture negated adjectives, like `(not, good)` and `(not, bad)`, which can be very impactful in applications such as sentiment analysis.

Bag-of-words models are a strong baseline for text classification, even though they disregard word order. If preserving word order is desirable, convolutional and recurrent neural network approaches can do so by taking as input the sequence of vocabulary entries and learning to construct appropriate representations.

2.4.3 Metrics

We use accuracy and F_1 score as metrics of classification performance.

Accuracy

Accuracy is the proportion of correct predictions:

$$\text{accuracy} = \frac{\#\text{ of correct predictions}}{\#\text{ of predictions}}\,. \tag{2.68}$$

In imbalanced datasets, in which one class is much more frequent, accuracy is not a good metric. In this cases, the majority class baseline—a classifier that always chooses the most frequent class—yields a high accuracy without even taking its input into consideration. The average F_1 score is more appropriate for such cases.

F_1 score

The F_1 score is defined as:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}\,, \tag{2.69}$$

whereas precision and recall are defined as follows: let tp, fp and fn be the number of true positives, false positives and false negatives, respectively. Then,

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}\,, \tag{2.70}$$

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}\,. \tag{2.71}$$

The F_1 score as defined is a measure of class classification performance: to evaluate the task as a whole, the scores for each class must be aggregated in some way. We report both average and weighted by class frequency F_1 scores, computed as follows:

$$\text{average } F_1 = \frac{1}{c}\sum_{i=1}^{c} F_{1i}\,, \tag{2.72}$$

$$\text{weighted } F_1 = \frac{1}{n} \sum_{i=1}^{c} n_i \cdot F_{1i} \,, \tag{2.73}$$

where c is the number of classes, F_{1i} is the F_1 score of class i, n_i is the number of class i samples and n is the total of samples.

To show why this metric is more adequate than accuracy for imbalanced datasets, we shall use as an example a binary classification problem where the most frequent class has a frequency of 90%. In this case, a majority baseline yields an accuracy of 0.9 and a majority class F_1 score of 0.9474, both pretty high values. However, the F_1 score for the minority class would be zero[9], resulting in average and weighted F_1 scores of 0.4737 and 0.8527 respectively.

To improve readability, in this dissertation we report metrics as percentages; e.g. 94.74 instead of 0.9474.

2.4.4 Transfer learning

Supervised learning requires labelled data, a scarce and expensive resource, even more so when compared with the wealth of natural text readily available online. Transfer learning techniques can be employed to leverage unlabelled data in a unsupervised way to improve models trained in a supervised way. In this dissertation we do so by using GloVe [105] word vectors pre-trained on Portuguese corpora [46] and training classifiers on top of pre-trained language models.

GloVe

Instead of using a bag-of-words model, which represents documents through a sparse vector of the size of the vocabulary, one can represent the document tokens as comparably low-dimensional dense vectors that capture word meanings. The main intuition is that the obtained vectors (also called embeddings) should cluster together words with similar meanings. This is done by processing unlabelled corpora and minimising a cost function.

The GloVe (Global Vectors) training procedure takes all possible pairs of vocabulary words and minimises the difference between the dot product of each pair's embeddings and the number of times they co-occur in the training corpus:

$$J = \sum_{i,j=1}^{\mathcal{V}} f(\mathbf{X}_{ij})(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \log \mathbf{X}_{ij})^2 \tag{2.74}$$

[9] Technically, since the minority class is never predicted, $tp + fp = 0$ and the precision is undefined. Therefore, the F_1 score of the minority class is undefined. For the sake of simplicity we consider the F_1 score to be 0 whenever recall or precision is 0, which makes intuitive sense.

where $f(\cdot)$ is a weighting function introduced to prevent overweighing frequent and rare co-occurrences, $\mathcal{V}$ is the vocabulary size, $\mathbf{w}_i$ are word embeddings, $\tilde{\mathbf{w}}_j$ are context word embeddings, b_i and $\tilde{b}_j$ are biases for word and context word embeddings and $\mathbf{X}$ is a matrix of co-occurrence counts. The final vector for word i is the sum of $\mathbf{w}_i$ and $\tilde{\mathbf{w}}_i$.

Language model pretraining

Language modelling is an unsupervised task whose objective is to predict the next word in a text sequence. To do this well a model needs to be able to identify language features involving syntax and semantics. As such, like pre-trained word embeddings, language models are able to learn general-purpose representations [119] to be used in downstream tasks. Language models have an advantage though, since their representations are contextual: the same word in different contexts will have different representations.

A deep neural network trained on a language modelling task can be used in downstream supervised tasks by attaching a linear classifier on top of it and training it on the target task data. We further detail this approach in Chapter 5.

Chapter 3

LeNER-Br dataset

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 chapter 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 [94]. This evaluation demonstrate that biLSTM-CRF outperforms the state of the art of that dataset. We then retrained biLSTM-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].

3.1 Introduction

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.

[1] An early version of this chapter has been published in: Pedro H. Luz de Araujo, 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 [87].

The state-of-the-art entity recognition systems [76, 90] 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 [92]. The process of labelling data is expensive and time consuming since the best corpora are manually tagged by humans.

Although state-of-the-art English NER models are approaching human performance, they do not generalise well to other domains [5]. 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.

There are few manually annotated corpora in Portuguese. Some examples are the first and second HAREM [124, 36] and Paramopama [94]. Another approach is to automatically tag a corpus, like the one proposed in [100] 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 [94].

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.

> 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. Our objectives are:

- to describe a novel dataset of legislative and court documents manually annotated with named entity labels;
- to train a biLSTM-CRF on the data to serve as a benchmark for future work;
- to train the same architecture on the Paramopama corpus [94] to assess the viability of the model for Portuguese datasets.

Our hypotheses are twofold:

1. a biLSTM-CRF model trained on Paramopama will push the state of the art for that corpus;
2. the classification results for the general entities in LeNER-Br (person, location organisation and time entities) will be comparable to the ones for the domain-specific entities (legislation and legal case entities). That is, there will be not a large discrepancy between general and specific entity classification metrics.

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 engineered contextual rules and lookup tables that are not automatically inferred through machine learning. Cardellino et al. [19] 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 (§3.2). We then present the model used to evaluate our dataset (§3.3), along with the training of the model and our choice of hyperparameters (§3.4). Following that, we present the results achieved regarding the test sets (§3.5) and our final considerations (§3.6).

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

3.2 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.

For each document, the NLTK [6] 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 were divided between two colleagues and me for annotation. WebAnno [27] was employed to manually annotate each 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 [52] respectively. Since I was the only annotator with legal training, I revised all documents to check for annotation correctness and consistency.

The IOB tagging scheme [111] 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 validation 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 [137] datasets (318,073, 310,000 and 301,418 tokens respectively). Table 3.1 presents the number of tokens and sentences of each set and Table 3.2 displays the number of words in named entities of each set per class. Table 3.3 presents an excerpt from the training set.

Table 3.1: Sentence, token and document count for each set.

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

Table 3.2: Named entity word count for each set.

Category	Training set	Validation 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

3.3 The baseline model: biLSTM-CRF

To establish a methodological baseline on our dataset, we chose the biLSTM-CRF model, proposed in [76]. This model is proven to be capable of achieving state-of-the-art performance on the English CoNLL-2003 test set [137] (an F_1 of 90.94). It also has readily available open-source implementations [39].

The architecture of the model consists of a Bidirectional [43] Long Short-Term Memory (LSTM) [51] followed by a Conditional Random Fields (CRF) [75] 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 [105] word embeddings pre-trained on a multi-genre corpus formed by both Brazilian and European Portuguese texts [46]. 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 3.1 presents an overview of this process.

To reduce overfitting and improve the generalisation capabilities of the model a dropout mask [134] 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 3.2 shows the main architecture of the model.

3.4 Experiments and hyperparameters setting

Here we present the methods employed to train the model and display the hyperparameters that achieved the best performance. We use Python 3 [142] as the programming language and the model is implemented using Tensorflow 1 [1].

Table 3.3: 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
falta	O
de	O
intervenção	O
do	O
Ministério	B-ORGANIZACAO
Público	I-ORGANIZACAO
nas	O
ações	O
em	O
que	O
deva	O
figurar	O
como	O
fiscal	O
da	O
lei	O
e	O
da	O
Constituição	B-LEGISLACAO
(	O
custus	O
legis	O
et	O
constituitionis	O
’	O
)	O
enseja	O
de	O
forma	O
inexorável	O
a	O
nulidade	O
do	O
processo	O
,	O
segundo	O
prescreve	O
o	O
artigo	B-LEGISLACAO
279	I-LEGISLACAO
...	...

TJMG	B-ORGANIZACAO
-	O
Apelação	B-JURISPRUDENCIA
Cível	I-JURISPRUDENCIA
1.0549.15.003028-2/003	I-JURISPRUDENCIA
,	O
Relator	O
(	O
a	O
)	O
:	O
Des	O
.	O
(	O
a	O
)	O
Otávio	B-PESSOA
Portes	I-PESSOA
,	O
16ª	B-ORGANIZACAO
CÂMARA	I-ORGANIZACAO
CÍVEL	I-ORGANIZACAO
,	O
julgamento	O
em	O
28/09/2017	B-TEMPO
,	O
publicação	O
da	O
súmula	O
em	O
06/10/2017	B-TEMPO
)	O
Assim	O
sendo	O
,	O
entendo	O
que	O
deve	O
ser	O
acolhida	O
...	...

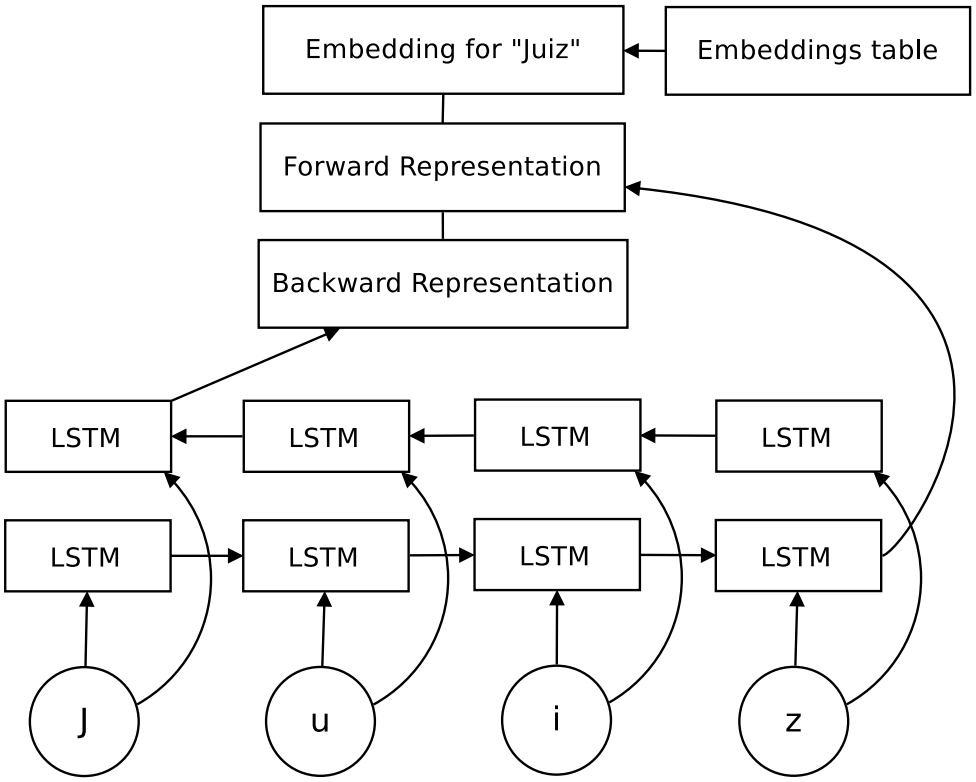


Figure 3.1: Each word vector representation is a result of the concatenation of the outputs of a bidirectional LSTM and the word level representation from the word lookup table.

Both Adam [70] 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 [76], presented in Table 3.4. It is worth noting that the number of LSTM units refers to one direction only. Since the LSTM layers 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 [94] 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 confirming that the model performed better than the state-of-the-art model (ParamopamaWNN [95]), the biLSTM-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.

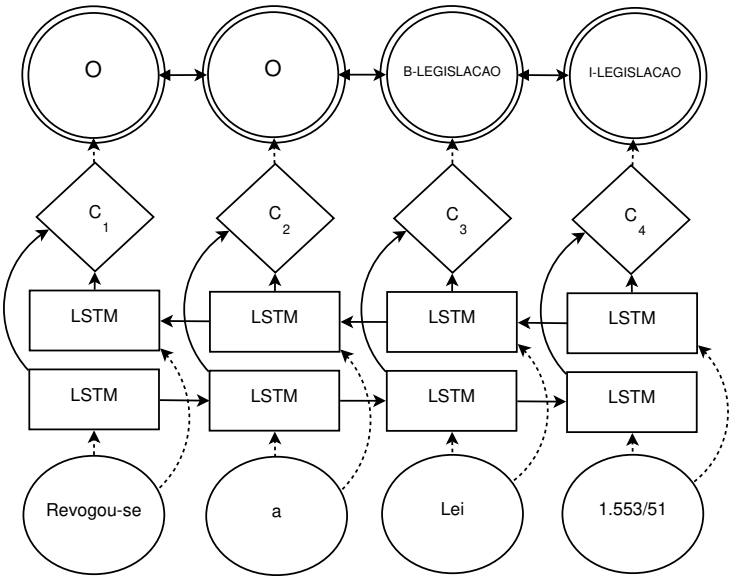


Figure 3.2: The biLSTM-CRF model. The word vector representations serve as input to a bidirectional LSTM layer. C_i represents the concatenation of left and right context of word i. Dotted lines represent connections after a dropout layer is applied.

Table 3.4: 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

3.5 Results

The metric used to evaluate the performance of the model on both datasets was the F_1 Score. Tables 3.5 and 3.6 compare the performance of the biLSTM-CRF [76] and ParamopamaWNN [95] models on different test sets. Test Set 1 and Test Set 2 are the last 10% of the WikiNER [100] and HAREM [124] corpora respectively. Table 3.7 shows the token prediction scores achieved by the biLSTM-CRF model when training on the proposed dataset, that is, correctness is assessed for each token individually. Table 3.8 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.

Table 3.5: Results (in %) on Paramopama Test Set 1 (10% of the WikiNER [100]) 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 3.6: Results (in %) on Paramopama Test Set 2 (HAREM [124]) 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 3.7: 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

The obtained results show that the biLSTM-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, confirming our first hypothesis (p. 31).

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. Despite that, the obtained results on LeNER-Br show that a model trained with it can

Table 3.8: 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

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%.

Furthermore, scores for the legal entities were comparable to the ones from the general entities, with average ($\pm$ standard deviation) entity classification scores of 88.02 ± 8.54 and 80.91 ± 9.94, respectively, confirming our second hypothesis (p. 31).

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.

3.6 Summary

We presented 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 biLSTM-CRF, trained on this dataset was able to achieve a good performance: weighted F_1 score of 92.53 and 86.61 for token and entity classification, respectively. There is room for improvement, which means that this dataset will be relevant to benchmark methods that are still 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.

3.7 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 present VICTOR, a dataset of documents from Brazil's Supreme Court, and the tasks we explored in that context: text and multimodal classification and topic modelling.

Chapter 4

VICTOR dataset

In this chapter we present VICTOR[1], a dataset of legal documents with manual annotation for two classification tasks: document type classification and lawsuit theme assignment. Section 4.1 presents the data and establishes benchmarks for each task using different methods for text representation and classifiers. Section 4.2 proposes the use of latent Dirchlet allocation (LDA) to model the dataset's lawsuits. We first assess the topics obtained by examining their semantics and labelling them. Then, we measure topic quality by using topic distribution vectors as input to a general repercussion theme classifier. Finally, Section 4.3 proposes a method that combines visual and textual features as well as sequential cues to improve document classification performance.

4.1 VICTOR: a dataset for Brazilian legal documents classification

This section describes VICTOR, a novel dataset built from Brazil's Supreme Court digitalized legal documents. It is 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

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

use all pages. Contrary to the Court experts' expectations, we find that using all available data is the better method[2].

4.1.1 Introduction

The Brazilian court system is burdened by a large number of lawsuits. In 2019, there were 77.1 million lawsuits awaiting judgment—almost one lawsuit for every three Brazilians. Some of these lawsuits will stay in the system for a long time, with average processing times that can reach more than six years. All of this contributes to raising the legal system cost: that same year, Brazil spent about R$100 billion in expenses with the judiciary, about 25 billion dollars considering the average exchange rate in 2019 [127].

Natural language processing (NLP) and machine learning (ML) techniques can improve this scenario by enabling faster and more efficient document analysis. Brazil's Supreme Court receives roughly 42 thousand cases each semester, which takes about 22 thousand hours for humans to sort through [136]. This time could be better spent on more complex stages of the workflow, such as those requiring legal reasoning.

The cases reach the court as mostly unstructured and unindexed PDF files of raster-scanned documents [86]. Therefore, as a first goal we explore and evaluate methods for automatically classifying document types. Intra-class diversity and document quality are the main challenges: the documents range from petitions and evidence to rulings and orders, originate from different Brazilian courts and often contain visual noise such as handwritten annotation, stamps, and stains (Figure 4.1).

In addition, lawsuits pertaining to the Brazil's Supreme Court—*Supremo Tribunal Federal* (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.

Thus, our objectives are:

- to describe a novel dataset of lawsuits from Brazil's Supreme Court annotated with document type and general repercussion theme labels;
- to train bag-of-words and deep models to serve as a benchmark for the two tasks.

Our hypothesis are:

1. leveraging the sequential aspect of lawsuits improves classification performance on the document type classification task;

[2]An early version of this section has been published in: P. H. Luz de Araujo, Teófilo E. de Campos, Fabricio Ataides Braz and Nilton Correia da Silva: VICTOR: a dataset for Brazilian legal documents classification [86].

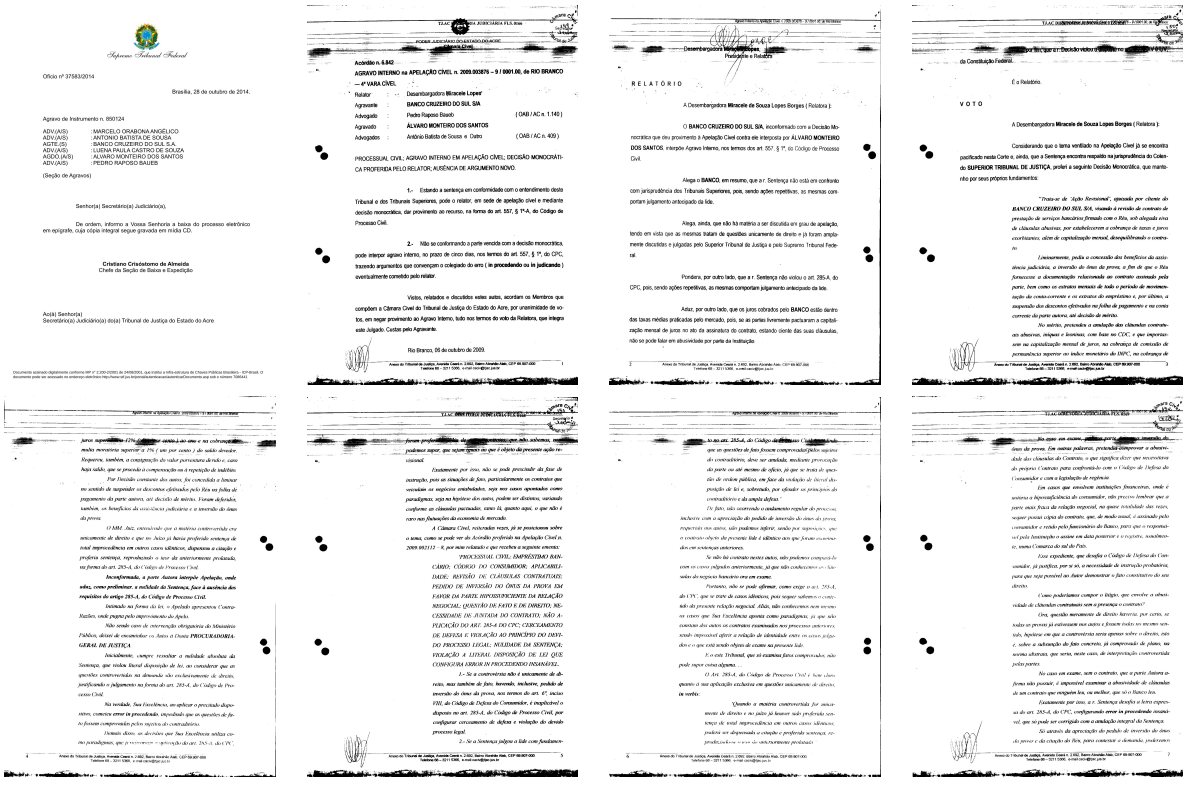


Figure 4.1: The first eight pages of a lawsuit. While the first page is clean, the others come from an older document and contain ink stains, stamps, handwritten signatures and other artifacts.

2. filtering out less informative pages will improve classification performance on the theme assignment task.

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[3].

The rest of this section is organised as follows. We first introduce other works related to text classification and processing of legal domain documents (§4.1.2). Then we discuss the dataset and its creation process (§4.1.3). We present the models explored and the experiments involved and discuss the results obtained regarding the first (§4.1.4) and second goals (§4.1.5), respectively. Finally, we conclude the work by presenting our final considerations (§4.1.6).

4.1.2 Related work

Text classification

Text classification is a 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 and developing classifiers [150]. Applications include spam filtering [30], sentiment analysis [2] and topic identification [144]. 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.

A traditional well-performing baseline for text classification is representing a document as a BOW and give that as input to a classifier like Naïve Bayes (NB) or Support Vector Machine (SVM) [63]. 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. [65] propose a shallow model that

[3]Resources and code) from this section are available at `https://cic.unb.br/~teodecampos/ViP/lrec/`.

uses n-gram features and hierarchical softmax to efficiently train on large datasets. Liu et al. [80] 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. [150] and Conneau et al. [24] independently show that a character-level CNN surpasses shallow models' performances on large datasets. Johnson and Zhang [64] were able to improve the state of the art by using a word-level LSTM network with pooling. Howard and Ruder [58] 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, 19, 87]. Automatic summarisation has been employed to help manage the great amount of information legal employees are required to process [66, 37, 74, 68]. In addition, topic models have been used to analyse large corpora of legal documents [20, 114, 102].

Text classification in the legal domain is used in a number of different applications. Katz et al. [67] 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. [3], 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. [135] 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. [139] 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.

4.1.3 The dataset

The VICTOR dataset is composed of 45,532 Extraordinary Appeals[4] (*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 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 [133] 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 applied 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)[5].

The data contains two types of annotation for two different tasks, as exemplified by Figure 4.2.

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 [14, 26].

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] Appeals on the grounds of conflit with constitutional law.

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

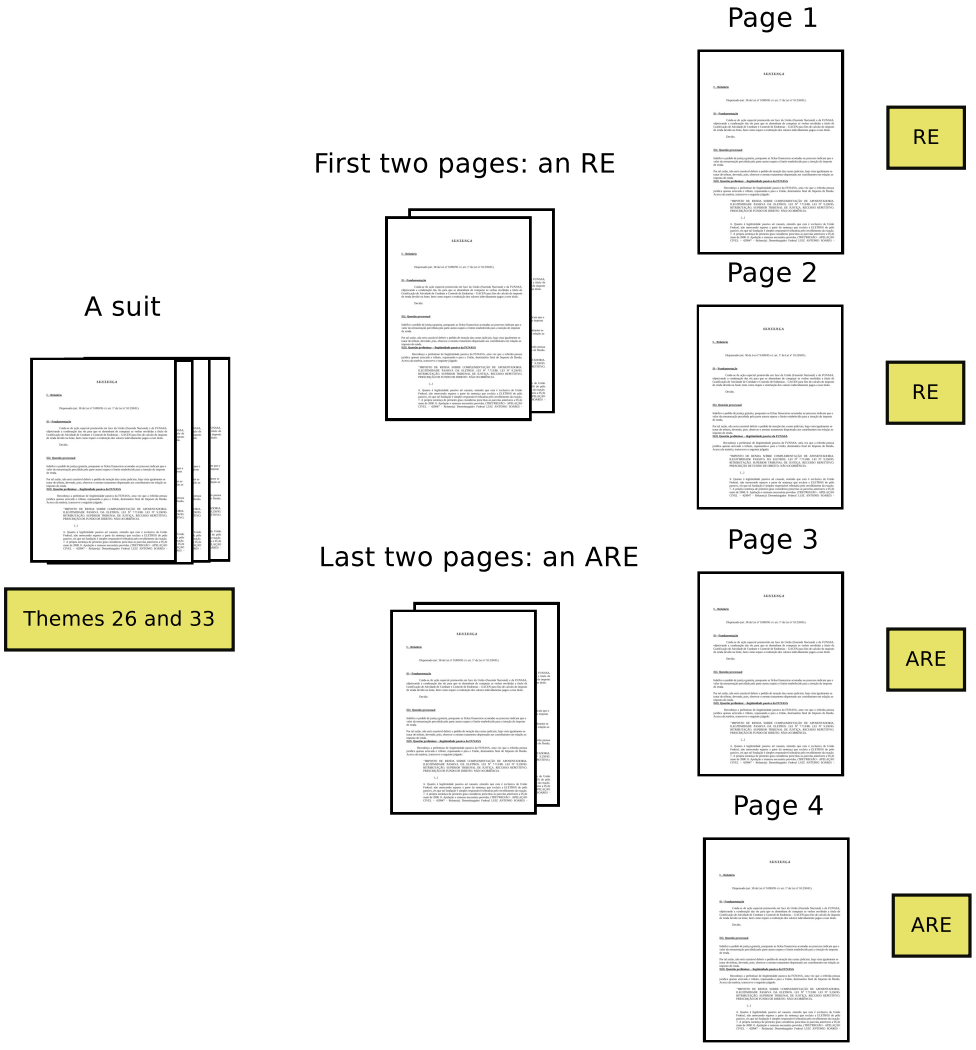


Figure 4.2: The two types of annotation, as exemplified by a suit of four pages. Each suit is assigned to one or more themes (left column) and is composed of one or more documents (middle column). The documents are in turn composed of one or more pages, each of them labelled with its document type (right column).

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, where the number of suits for each theme is capped at 100 samples in each set, resulting in 6,510 Extraordinary Appeals, 94,267 documents and 339,478 pages.

Table 4.1 exhibits the document type distribution for each split of the relevant versions of the dataset. Figures 4.3, 4.4 and 4.5 show the theme distribution for each versions of VICTOR. The presented theme IDs are the ones originally used by the Court[6].

Table 4.1: 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

4.1.4 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

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

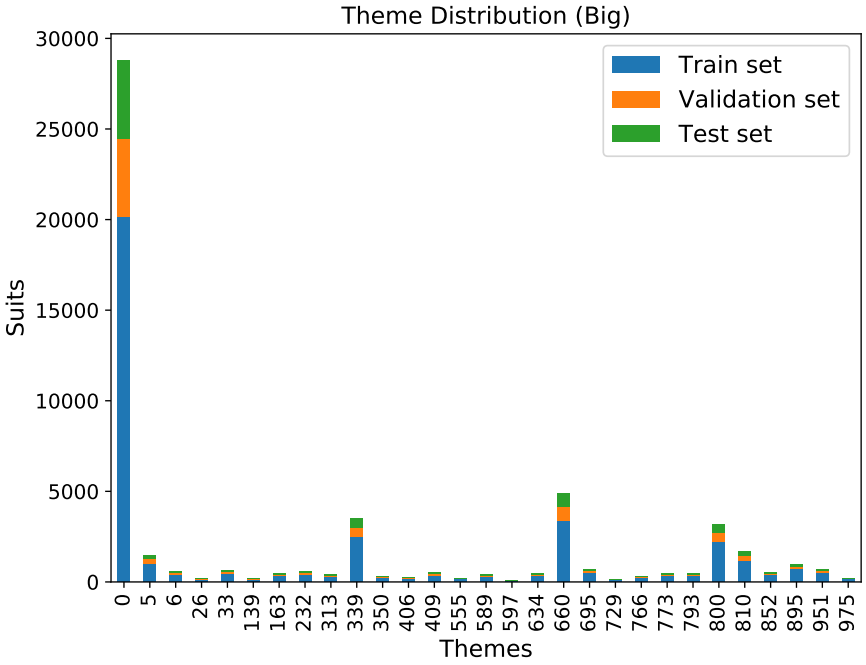


Figure 4.3: BVic theme distribution.

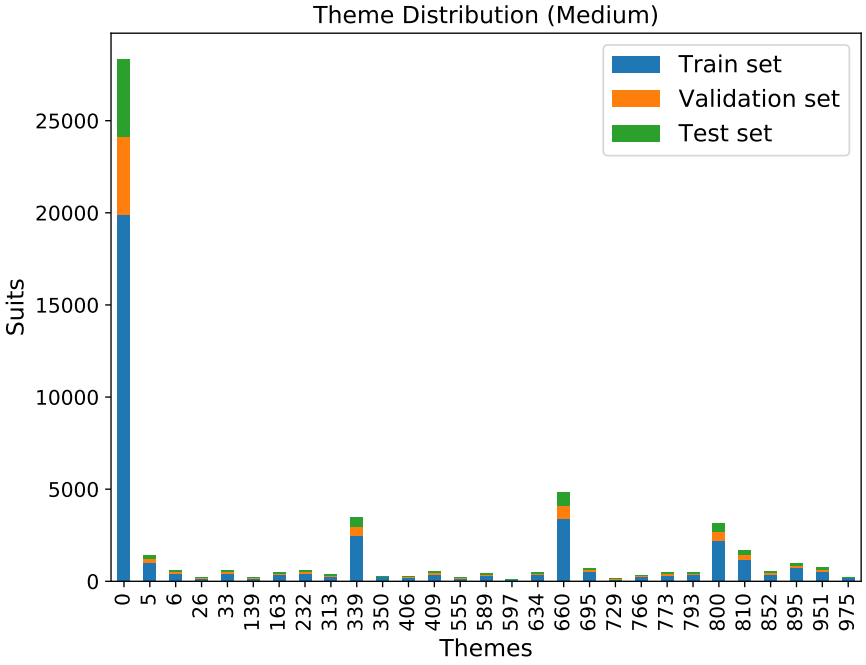


Figure 4.4: MVic theme distribution.

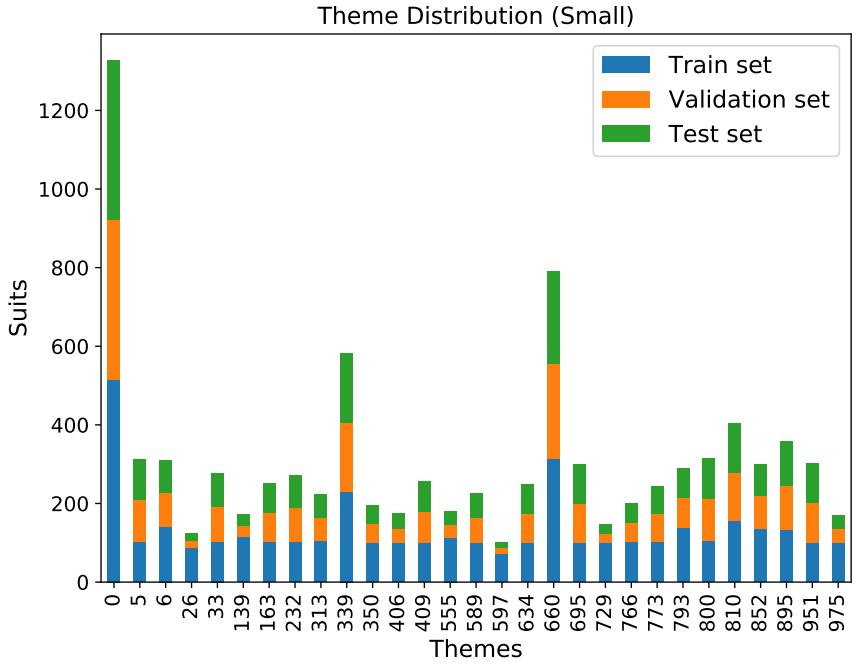


Figure 4.5: SVic theme distribution.

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[7].

Feature extraction: We use random search to choose the best hyperparameters and evaluating on 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. Let c be the number of classes and $\mathbf{w}$ a c-dimensional vector whose component i is the weight for class i. Then the weights are computed by the following equation, as implemented in the scikit-learn library [18]:

[7]We use the scikit-learn library [17] to train and evaluate the BOW models.

$$w_i = \frac{n}{c \cdot f_i}\,, \tag{4.1}$$

where n is the number of training samples and f_i is the number of samples from class i.

Convolutional Neural Network

We based our CNN architecture[8] on the one proposed in [24]. 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 4.6. 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 [70] to optimise 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 [43] Long Short-Term Memory (LSTM) [51] 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 4.7 exhibits the architecture.

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

[8] We use the Keras library [23] to train and evaluate the CNN and LSTM models.

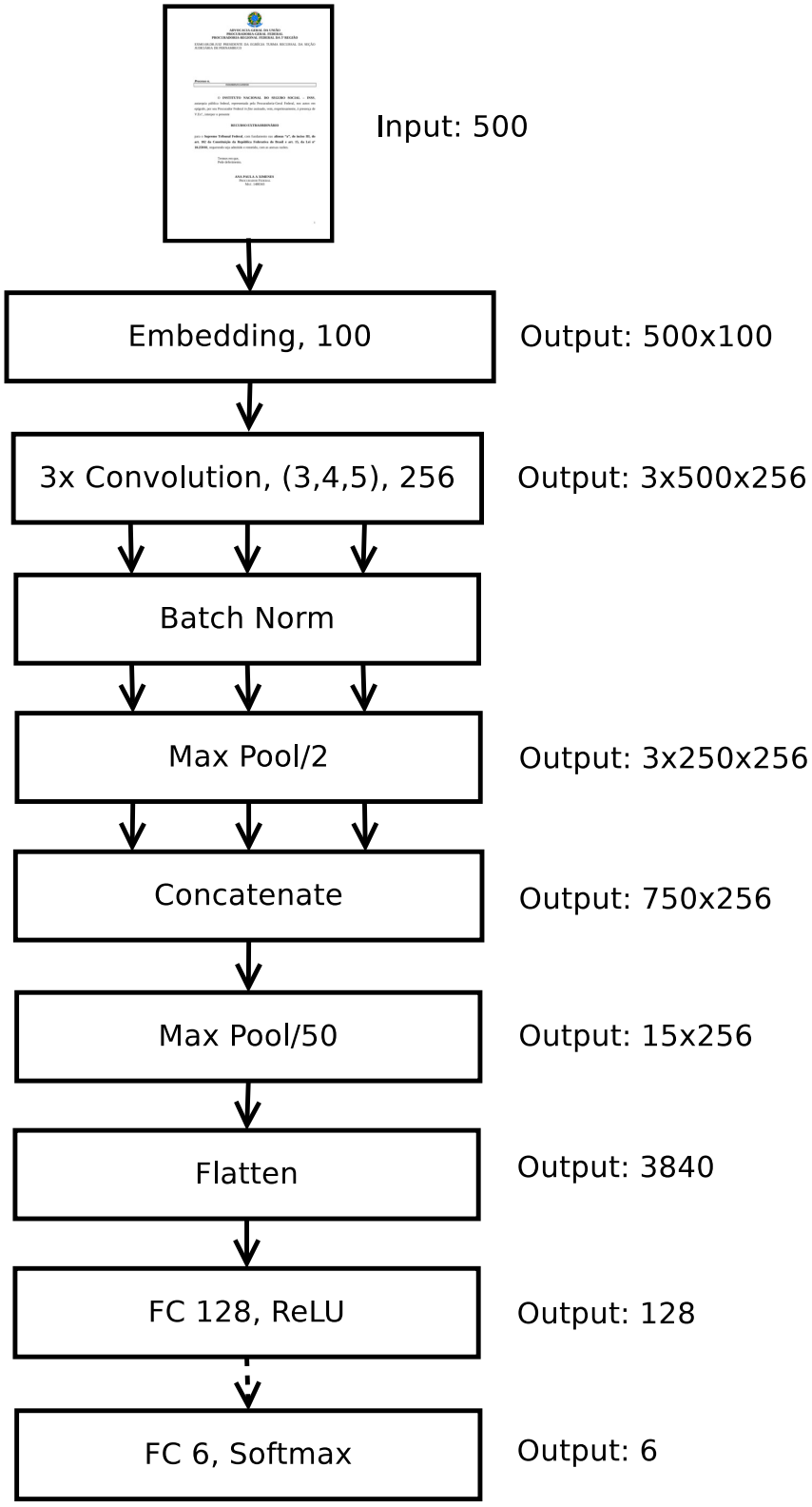


Figure 4.6: CNN architecture for document type classification. The dashed line indicates dropout was applied.

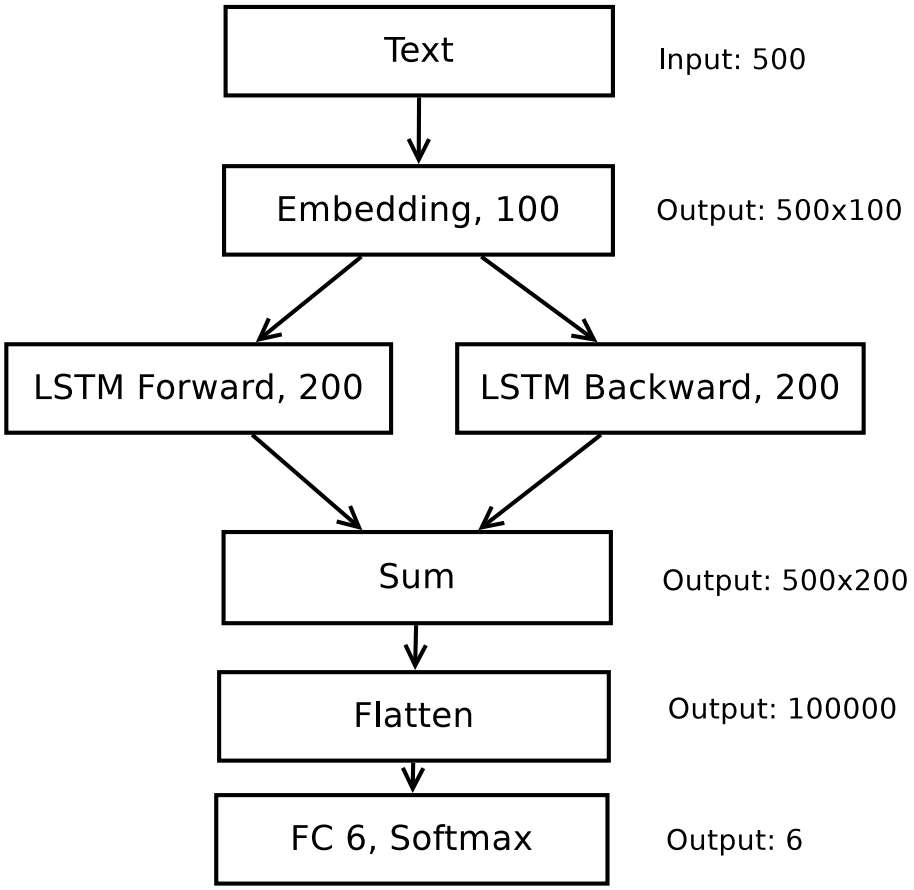


Figure 4.7: Bi-LSTM architecture for document type classification.

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 [9], which have been shown to be very effective in sequence tagging problems [75, 59, 76].

To better leverage the sequential information, we adapt the document classes by using the IOB tagging scheme [111]. 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 4.2 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.

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

[9] We use the sklearn-crfsuite library [72] to train the CRF model.

Table 4.2: 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

behaved approximately as the baseline, predicting almost every sample as belonging to the *Others* class.

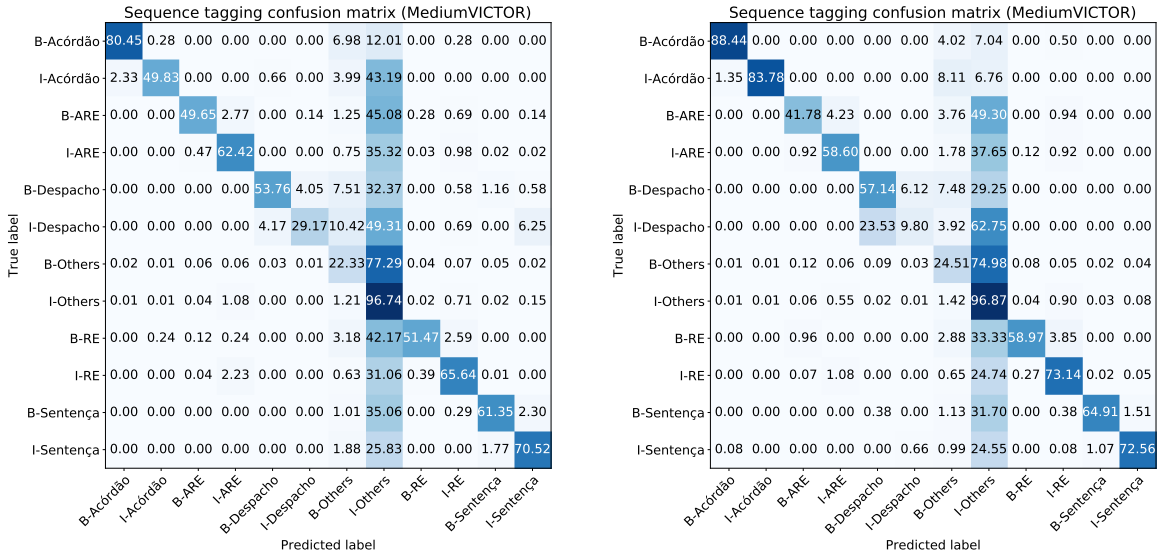
Table 4.3 shows the impact of CRF modelling. Our sequence modelling approach, albeit simple, results in overall improvements in both versions of dataset. The best increase in performance was regarding *Despacho* classification on MVic—a relative improvement of 11.62%. On the other hand, SVic's *Despacho* 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 4.3: 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%

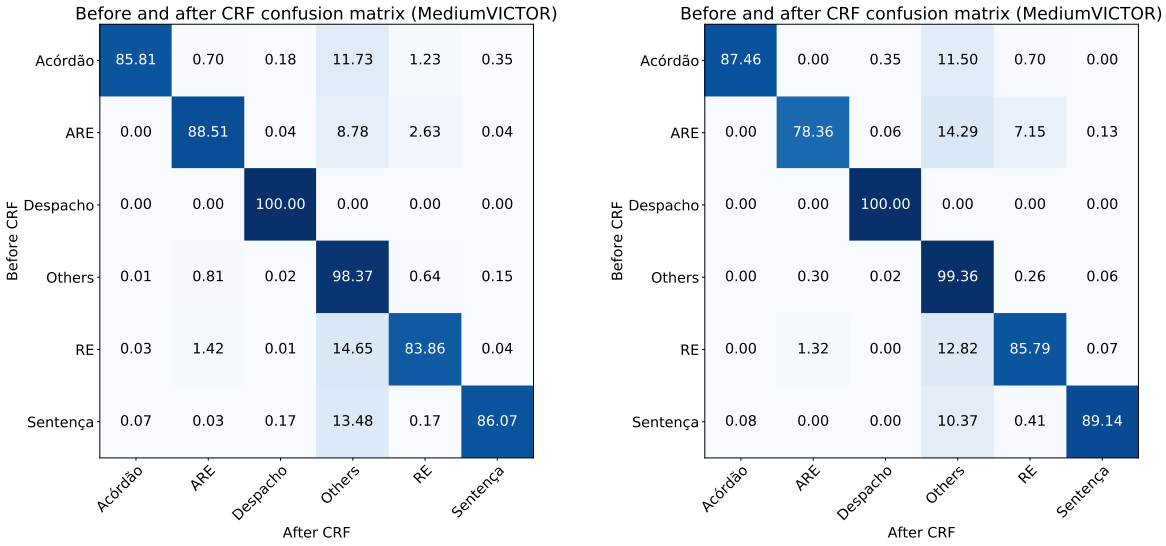
Figure 4.8 exhibits the confusion matrices of CRF tag predictions. The greatest source of confusion is the I-Others tag (pages classified as being inside of a "Others" document), which is not surprising due to its overabundance. We have a similar scenario when we analyse the confusion between predictions before and after CRF processing (Figure 4.9): the CRF is more likely to tag a page as *Others* when compared to the original model.

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



(a) MVic. (b) Svic.

Figure 4.8: 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.



(a) MVic. (b) Svic.

Figure 4.9: 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.

where we jointly train the CRF layer and the feature extractor[10]. 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. We further explore sequential modelling of pages in Section 4.3.

4.1.5 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 Boosting (XGBoost) [22] classifier. XGBoost is an optmised tree boosting system that has become very popular amongst Kaggle[11] competitions for various ML tasks.

Since theme classification is a multi-label and multi-class problem we employ an One-vs-All 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:

$$\forall c \in C, \text{ we assign } c \text{ to } d \text{ if } f_c(d) \geq t\,. \tag{4.2}$$

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 4.1.4.

XGBoost: We train 500 trees with a maximum depth of 4 and a shrinkage factor[12] of 0.1.

[10] An approach similar to the one described in Chapter 3.

[11] 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.

[12] Regularization factor that scales the contribution of each tree [47, p. 364].

Theme classification with domain knowledge

Humans tend to examine only the more informative documents when assigning themes to a suit. On that premise, domain experts have the intuition that a theme classification model can ignore less informative pages—the ones labelled as *Others*.

In order to test for this, one would have to filter out such pages from the data prior to training and evaluating a model. 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*.

Results and discussion

Table 4.4 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 4.5 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 4.4: 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 4.5: 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**	

4.1.6 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. On the other hand, filtering out pages classified as *others* did not improve the performance of theme assignment. 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 report our work on topic modelling VICTOR's lawsuits.

4.2 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 §4.1.3, which contains 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 quantitatively evaluate their potential using the topics to generate feature vectors for each appeal. These vectors are then used to train a lawsuit theme classifier. We compare traditional bag-of-words approaches (word counts and tf-idf values) with the topic-based text representation to assess topic relevancy. 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 vectoriser and that the interpretable, low dimensional representations it generates achieve good classification results[13].

[13] An early version of this section has been published in: P. H. Luz de Araujo and Teófilo E. de Campos: Topic Modelling Brazilian Supreme Court Lawsuits [85].

4.2.1 Introduction

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 [7]. 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 [93] and document classification [116].

In this section, we employ latent Dirichlet allocation (LDA) to model Big VICTOR's lawsuits. We measure topic quality in two ways. First, we manually inspect each topic and label it according to our interpretation of their semantics. Then, to measure how relevant the obtained topics are to matters important to the Court, we use them as input to a general theme classifier and use the classification performance as a quantitative proxy of topic quality[14]. Our objectives are:

- to model Brazil's Supreme Court lawsuits using latent Dirichlet allocation (LDA);
- to manually inspect obtained topics to interpret their semantics;
- to use topic distribution vectors as input to a classification task and evaluate the obtained model's performance.

Our hypotheses are twofold:

1. the obtained topics will relate to legal matters;
2. the trained models will outperform a baseline that assigns all themes to all suits.

Though some works already explore the use of artificial intelligence in the context of Brazil's courts [26, 87, 28], we are not aware of publications regarding the topic modelling of Brazilian lawsuits. Our contributions are:

1. The qualitative analysis of the semantics of each topic from models with 10 and 30 topics trained on the STF data.
2. The quantitative analysis of topic relevance by using topic distribution vectors as input for general repercussion theme classification. We experiment with models of 10, 30, 100, 300 and 1,000 topics.

[14]Resources (data and code) from this section are available at `https://cic.unb.br/~teodecampos/ViP/jurix2020/`.

The rest of the section is organised as follows. First, we briefly review topic model literature and NLP applied to the legal domain approaches (§4.2.2). Then we describe the model employed (§4.2.3). Following that, we report our experiments (§4.2.4) and present and discuss the results (§4.2.5). Last, we present our final considerations (§4.2.6).

4.2.2 Related work

Topic models

Topic models have been an area of research since 1990, when Deerwester et al. [29] proposed latent semantic indexing (LSI). The method uses singular value decomposition (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 [55], who introduced probabilistic LSI (PLSI). Like LSI, PLSI decomposes a co-occurrence matrix, but while the former uses a linear algebra approach, the latter method is statistical, modelling 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. [9] proposed latent Dirichlet allocation (LDA). The authors show that LDA can be used for a range of tasks, such as document modelling, 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 [7]. For example, by relaxing the assumption that the order of the documents can be neglected, Blei and Lafferty [8] propose Dynamic Topic Models, capable of modelling 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. NLP has been used to automatically extract and classify relevant entities in court documents [32, 19, 87]. Other works [66, 37, 74, 68] focus on using automatic summarisation to reduce the amount of information legal professionals have to process. Document classification has been explored for decision prediction [3, 67], area of legal practice attribution [135] and fine-grained legal-issue classification [139].

LDA has been employed to model legal corpora. Carter et al. [20] model documents from the Australian High Court; Remmits [114] models decisions from the Supreme Court of the Netherlands; O'Neill et al. [102] used LDA to explore British legislative texts.

Some works explore the processing of Brazilian legal documents. Correia da Silva et al. [26] use a CNN to classify STF's documents. De Vargas Feijó and Moreira [28] introduce a dataset for decision summarisation. Luz de Araujo et al. [87] 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 examining topic modelling of Brazilian legal corpora.

4.2.3 The model

Inspired by previous attempts to model different kinds of legal text [20, 114, 102], we choose LDA [9] as the method for topic generation. We use the following terminology [9]:

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

4.2.4 Experiments

Model training for exploratory analysis

We perform an exploratory analysis of the data aiming to understand its most relevant topics by training LDA models. 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 [54] for the online training of LDA models[15], based on stochastic optimisation 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 have a quantitative analysis of the detected topics, 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) [22]—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 the same One-vs-All approach described in Section 4.1.5.

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.

[15] As implemented in the Gensim library [113].

4.2.5 Results

Topic analysis

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 [44]. Tables 4.6 and 4.7 present the results of the labelling process. For each topic we show its four most relevant words, where relevance is defined [128] as

$$r(\mathbf{w}, \mathbf{z}|\lambda) = \lambda \log P(\mathbf{w}|\mathbf{z}) + (1 - \lambda) \log \frac{P(\mathbf{w}|\mathbf{z})}{P(\mathbf{w})}, \tag{4.3}$$

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[16], 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 4.6: 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*, *bruyn*
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.

[16] We use the pyLDAvis library [128] for topic visualisation.

Table 4.7: Topic labels and their respective four most relevant words (10 topics) in the original language.

Topic	λ	Assigned label	Words
1	0,6	Remuneração de servidor público	servidores, servidor, prescrição, remuneração
2	0	Direito Penal	entorpecente, hidrômetro, clandestino, interrogatório,
3	0,6	Direito Previdenciário	benefício, evento, aposentadoria, previdenciário,
4	0,6	Direito Civil	banco, contrato, consumidor, projudi
5	0,6	Direito à Saúde	saúde, município, municipal, medicamentos
6	0,4	Erros de OCR	ento, não, ro, co
7	0,6	Direito Tributário	icms, ipi, imposto, receita
8	0	Entidades	econorte, rcte, pieter, bruyn
9	0,4	Questões trabalhistas	fgts, pss, horas, folha
10	0,6	Publicidade do processo	original, site, acesse, informe

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 Tables 4.8and 4.9, 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 the most relevant words from each topic 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).

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

Figure 4.10 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

Table 4.8: 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

Table 4.9: Topic labels and their respective four most relevant words (30 topics) in the original language.

Topic	λ	Assigned label	Words
1	0,6	Responsabilidade Civil	dano, danos, indenização, moral
2	0,22	Decadência benefício previdenciário	benefício, decadência, teto, previdenciário
3	0,6	Direito Tributário	fazenda, tributário, receita, imposto
4	0,1	Miscelânea - Vocabulário jurídico, entidades e leis	nºsérie, pet, carimbo, itaperuna
5	0,4	Gratificação de servidores públicos	gratificação, desempenho, inativos, avaliação
6	0,4	Previdência rural	rural, contribuição, LEI_8212, aposentadoria
7	0,6	Reajuste de vencimento de servidor	reajuste, servidores, vencimentos, urv
8	0,4	Erros de OCR	ento, não, ro, ffl
9	0,6	Servidores militares	militar, servidor, militares, servidores
10	0	Direito Penal	clandestino, sepetiba, semiaberto, entorpecente
11	0,4	Direito Contratual/Comercial	contrato, contratos, taxa, contas
12	0,05	Conselhos técnicos	confea, crea, agronomia, LEI_6496
13	0,2	Concursos públicos	concurso, candidato, edital, vagas
14	0,4	Antecipação de reajuste de vencimento	upag, pccs, trabalhista, LEI_8460
15	0,6	Direito à Saúde	saúde, medicamentos, tratamento, medicamento
16	0,9	Poupança, juros e correção monetária	correção, monetária, poupança, mora
17	0,6	Publicidade do processo	original, site, acesse, informe
18	0,6	Reclamações trabalhistas	estran, tst, entidade, reclamante
19	0,4	Miscelânea- Direito do Consumidor e Bahia	consumidor, salvador, bahia, pdf
20	0	Entidades - nomes	lauxen, tainá, heloise, soeli
21	0,7	Qualificações	num, normal, internamento, foz
22	0,5	Seguros	seguro, previd, instituto, dpu
23	0,4	Folhas de pagamento	horas, fgts, folha, extras
24	0	Miscelânea-Assembleias, estatutos e palavras estrangeiras	andaterra, peixer, funds, market
25	0,5	Documentos fiscais	ltda, ipi, nfe, icms
26	0,4	Processos relacionados ao Rio Grande do Sul	sul, grande, alegre, paese
27	0,4	Imposto de Renda	atualizado, meses, rra, irpf
28	0,2	Direito tributário-Circulação de mercadorias	compatível, issqn, saída, eireli
29	0,2	Miscelânea-Movimentação processual e Paraná	paraná, arq, curitiva, mov
30	0,4	Pagamentos	jam, vlr, recolhido, crédito

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.

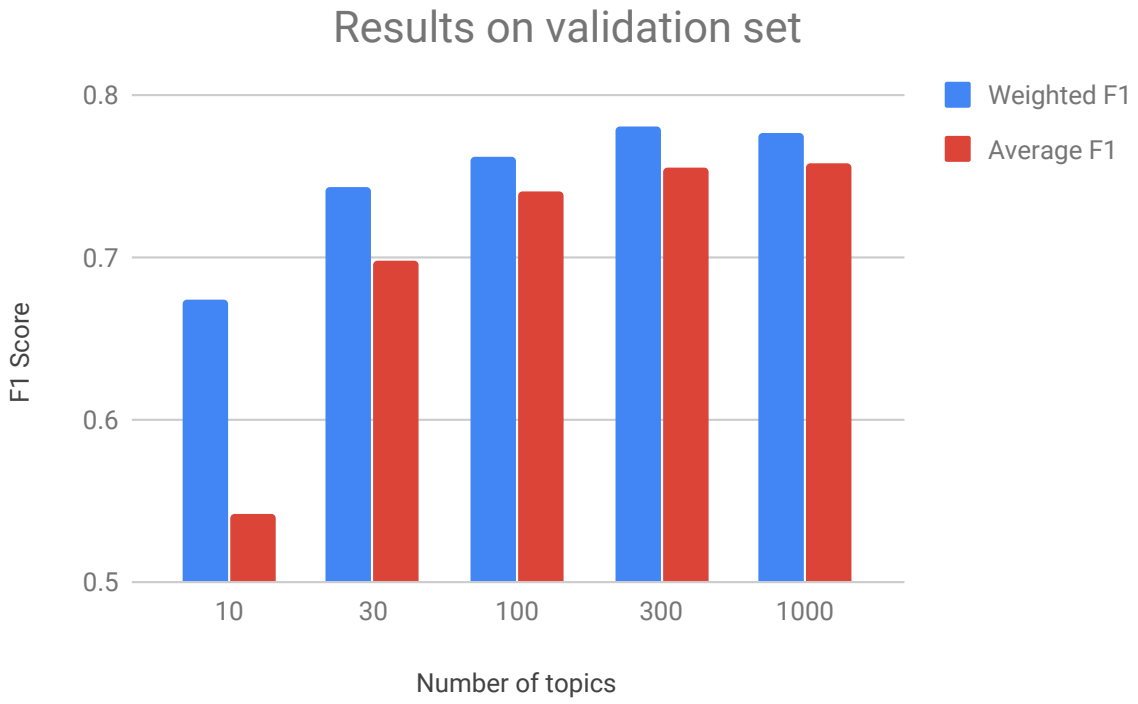


Figure 4.10: Validation set performance of classifiers trained with different numbers of topics. A baseline method that assigns all themes to any input achieves an F_1 score weighted by class frequency of 42.53 and an average F_1 score of 5.57.

Table 4.10 compares the 300-dimensional lawsuit representation with the word counts and tf-idf values bag-of-words representations on the test set. The topic distribution representation did not outperform the traditional methods, but achieved good performance—much better than the baseline that assigns all themes. These results suggest that the detected topics are related to the themes relevant to the Court and have the potential to aid the judiciary with the management of cases.

Furthermore, 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.

4.2.6 Summary

We used 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

Table 4.10: 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

between them and legal matters that reach the Court. We used the obtained topic distribution vectors as input for a supervised multi-label classification task in order to establish a quantitative analysis of topic relevance. 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 organise, explore and extract information of the massive amounts of data that reach the Court.

In the next section, we will expand SVic to include document images and propose a combination method that leverages visual, textual and sequential data.

4.3 Sequence-aware multimodal page classification of Brazilian legal documents

The Brazilian Supreme Court receives tens of thousands of cases each semester. Court employees spend thousands of hours to execute the initial analysis and classification of those cases—which takes effort away from posterior, more complex stages of the case management workflow. In this work, we explore multimodal classification of documents from Brazil's Supreme Court. We train and evaluate our methods on a novel multimodal dataset of 6,510 lawsuits (339,478 pages) with manual annotation assigning each page to one of six classes. Each lawsuit is an ordered sequence of pages, which are stored both as an image and as a corresponding text extracted through optical character recognition. We first train two unimodal classifiers: a ResNet pre-trained on ImageNet is fine-tuned on the images, and a convolutional network with filters of multiple kernel sizes is trained from scratch on document texts. We use them as extractors of visual and textual features, which are then combined through our proposed Fusion Module. Our Fusion Module can handle missing textual or visual input by using learned embeddings for missing data. Moreover, we experiment with bi-directional Long Short-Term Memory (biLSTM) networks and linear-chain conditional random fields to model the sequential nature of the pages. The multimodal approaches outperform both textual and visual classifiers, especially when leveraging the sequential nature of the pages[17].

4.3.1 Introduction

In Section 4.1 we explored page classification of legal documents using OCR-extracted text. However, document images may also contain useful discriminative features that would improve classification performance. In this section we extend SVic to include, in addition

[17]This section is based on a paper submitted to the International Journal on Document Analysis and Recognition (IJDAR).

to text, document images. Then, we explore and evaluate methods that automatically classify document pages by combining different sources of information. Our objectives are:

- to describe an expanded version of the SVic dataset that includes document images;
- to train and evaluate models that combine textual, visual and sequential information;

Our hypotheses are twofold:

1. leveraging visual input will improve classification performance when comparing with purely textual models;
2. leveraging sequential cues will improve classification performance of multimodal models.

Though previous work [86, 98] has examined Brazilian legal document classification, to the best of our knowledge we are the first to combine visual, textual, and sequential data to train better performing models. Our main contributions are:

1. a multimodal dataset of lawsuits composed of ordered document images and corresponding texts.
2. proposing and evaluating two multimodal combination methods and two sequence learning methods that leverage textual, visual and sequential information to improve Supreme Court document classification.
3. outperforming the state-of-the-art results on the small version of the VICTOR dataset [86].

The rest of this section is organised as follows. First, we examine previous work on multimodal document classification (§4.3.2). Then, we describe the data (§4.3.3) we used to train and evaluate our models. Following that, we detail our methods and experimental settings (§4.3.4). Finally, we discuss the results (§4.3.5) and conclude the paper (§4.3.6).

4.3.2 Related work

Textual and visual content are two of the four document aspects listed by Chen and Blostein [21] as possible feature sources (the other two are the physical layout and the logical structure). Image features range from fixed descriptors such as pixel density at different locations and scales [121] to approaches based on convolutional neural networks [61, 34, 4, 145, 98] such as VGG-16 [130] and MobileNetV2 [123]. Text features range from traditional methods such as latent semantic analysis [29] to pre-trained word embeddings (e.g. Fasttext [10]) and deep learning approaches [34, 4, 145].

These feature modalities may be used by themselves or combined to improve classification performance. This can happen at the feature level (early fusion) or at the prediction level (late fusion). Rusinol et al. [121] experiment with both options, trying different methods to combine predicted probabilities for late fusion (summing, multiplying, taking the maximum and logistic regression). Jain and Wigington [61] compare a spatially-aware early fusion method with four alternative methods of feature combination: concatenation, addition, compact bilinear pooling and gated units. The spatially-aware fusion underperformed simple feature combination, with concatenation, addition, and bilinear approaches performing similarly. Engin et al. [34] explore late and early fusion for the classification of Turkish banking documents, finding that both outperform unimodal methods. Mota et al. [98] investigate multimodal classification of Brazilian court documents, concluding that multimodal approaches compare favourably with unimodal ones.

Fewer works have explored incorporating sequential information. Rusinol et al. [121] use an n-gram model of the page stream that conditions page predictions on the types of the $n-1$ previous pages to capture their sequential nature. Wiedemann and Heyer [145] use as a feature of the target page the encoding of its predecessor. We described in §4.1.4 how we fed the predictions of a text classifier to a linear-chain conditional random field (CRF) to jointly predict pages of lawsuits.

In this work, we focus on the sequentially-aware early fusion of visual and textual features for legal document classification. To the best of our knowledge, this is the first work that considers both visual and textual modalities and sequential dependencies when classifying documents from the legal domain and in Portuguese—Luz de Araujo et al. [86] do not use visual features, while Mota et al. [98] do not leverage sequential information.

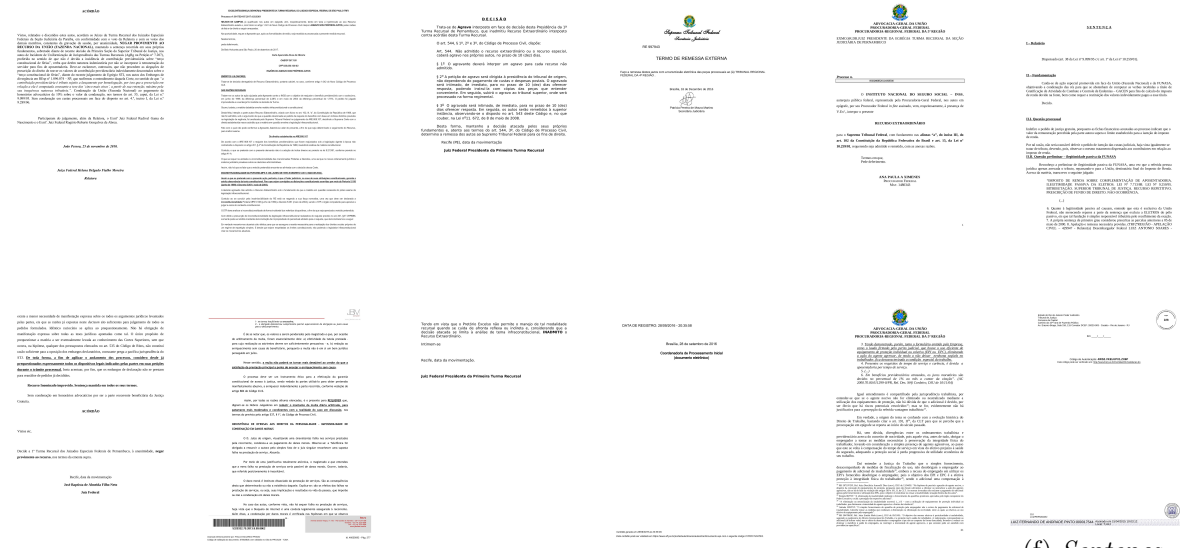
4.3.3 Data

We perform our experiments on an extension of SVic (§4.1.3), which we expanded to include, in addition to textual data, the document images. Every page in the expanded corpus is stored in at least one of two formats. First, as text extracted through optical character recognition [133], as we previously described (§4.1.3). Second, as JPEG images extracted from the original PDF files, with mean width and height of 1664 and 2322 pixels respectively.

Most of the samples contain both textual and visual sources of information, except for 33,849 images with no corresponding text and 4 texts with no corresponding image. We discuss our strategy for dealing with missing data when training fusion models in Section 4.3.4. Table 4.11 presents the number of text and image samples across data splits and classes. Due to the nature of the data, a document may appear more than once in a

Table 4.11: Class counts per split, showing the number of page images and text extracted through OCR. Between parentheses, the deduplicated counts.

Class	Training set		Validation set		Test set	
	Image	Text	Image	Text	Image	Text
Acórdão	583 (583)	553 (553)	320 (314)	299 (293)	287 (285)	273 (271)
ARE	4,258 (4,220)	2,546 (2,508)	2,798 (2,650)	2,149 (2,001)	2,655 (2,537)	1,841 (1,723)
Despacho	361 (361)	346 (346)	189 (183)	183 (177)	199 (198)	198 (197)
Others	144,583 (140,786)	134,134 (130,337)	95,602 (91,434)	84,104 (79,936)	92,529 (87,902)	85,408 (80,781)
RE	10,225 (10,181)	9,509 (9,465)	6,987 (6,803)	6,364 (6,180)	6,386 (6,177)	6,331 (6,122)
Sentença	2,177 (2,177)	2,129 (2,129)	1,681 (1,613)	1,636 (1,568)	1,503 (1,478)	1,475 (1,450)



(a) *Acórdão* (b) ARE (c) *Despacho* (d) Others (e) *Petição* (f) *Sentença*

Figure 4.11: First page (top row) and interior page samples for each class.

lawsuit, so we present both raw and deduplicated counts. That said, given that the corpus has been split by lawsuits, there is no sample intersection between splits.

Human agents find the first page of documents easier to classify when compared to interior pages. This is true considering both visual and textual aspects, since first pages contain highly informative cues, such as headers and titles. Figure 4.11 compares first page and interior page samples for each class. We validate this intuition in Section 4.3.5.

4.3.4 Methods

In this section we describe our methods for visual and textual page classification, feature fusion and sequence learning. We also describe the corresponding experimental settings. With the exception of the text classifier (implemented using the Keras [23] library), all methods are implementend using PyTorch [104] and FastAI [56]. Unless stated otherwise, we optimise the cross-entropy loss using Adam [70] and mini-batches of 64 samples. As metrics, we report arithmetic (average) and weighted by class frequencies (weighted) means of the F_1 score. We evaluate the models using the parameters with the best average

F_1 score computed on the validation set. That is, after each epoch, we only save model parameters if the validation performance is the highest up to that point.

Text Classification

We use the CNN architecture described in Section 4.1.4 as the method for text classification.

As a strategy to deal with class imbalance, we train a variant of the CNN model, which we call CNN-w, that weighs each sample contribution to the loss by a factor inversely proportional to its class frequency (Eq. 4.1).

Image Classification

To classify document images, we fine-tune a ResNet50 [48] model pretrained on ImageNet [122]. We first train only the head of the model for one epoch, employing a cyclic learning rate with cosine annealing [132]. Then, we train all layers for one cycle of 6 epochs with discriminative fine-tuning [57]. As we did for the text classifier, we train a variant of the model with factors inversely proportional to class frequency: ResNet50-w.

To choose learning rates, we use the learning rate range test [131]. That is, we train the model for a few iterations, starting from a low learning rate value and increasing it after each mini-batch, plotting the loss against the learning rates. We then pick a learning rate close to the point where the loss starts to increase—high enough for quick learning, but not so high as to impede learning.

Image and Text Combination Strategies

In this section we describe our proposed method for early fusion of visual and textual features, a baseline method for comparison and an ablation analysis of the fusion classifier.

Hybrid Classifier As a baseline method that fuses visual and textual data we use a hybrid classifier (HC) that works as follows: if textual data is available, use the best text classifier; otherwise, use the best image classifier. The intuition is that this approach would be at least as good as using only text data, which better discriminates the document classes when compared with visual data. Figure 4.12 illustrates the method.

Fusion Module To fuse visual and textual data we first compute representations using the trained text and image classifiers. As text embeddings, we extract the 3,840-dimensional activations of the flatten layer of the CNN (Fig. 4.6). As image embeddings, we take the activations of the last convolutional block in the ResNet and apply global average and global

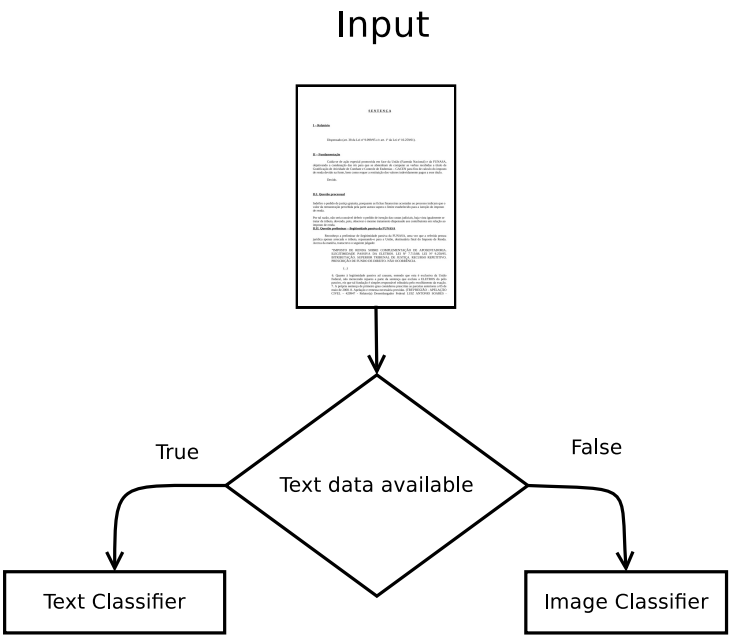


Figure 4.12: The hybrid classifier (HC): a baseline fusion classifier that only uses visual information if text data is not available.

max pooling. Then we concatenate and flatten the result, obtaining 4,096-dimensional vectors.

The pre-computed representations are concatenated and fed to a batch normalisation layer [60], followed by an FC layer with d units, batch normalisation, and a final FC layer. The softmax function produces the predictions. Figure 4.13 illustrates our proposed Fusion Module (FM).

In case of missing data, when only the document image or text is available—but not both—we experiment with two options. The first uses learnable embeddings for missing text or image; the other, simply uses a vector of zeroes in such cases (FM-zero).

We run preliminary experiments with one cycle of 10 epochs for each of four configurations, varying the number of hidden units d (128 or 512) and the use of learnable embeddings or zero vectors for missing data. We then train the model that obtained the highest average F_1 score from scratch for one cycle of 20 epochs. Learning rates are chosen using the range test.

Sequence Classification

Given that a lawsuit is composed of an ordered series of pages, one can, instead of classifying each page by itself, leverage the sequential nature of the data by treating the problem as a sequence labelling task. That is, rather than having a page and a class prediction as input and output, the sequence classification approach outputs a sequence of class

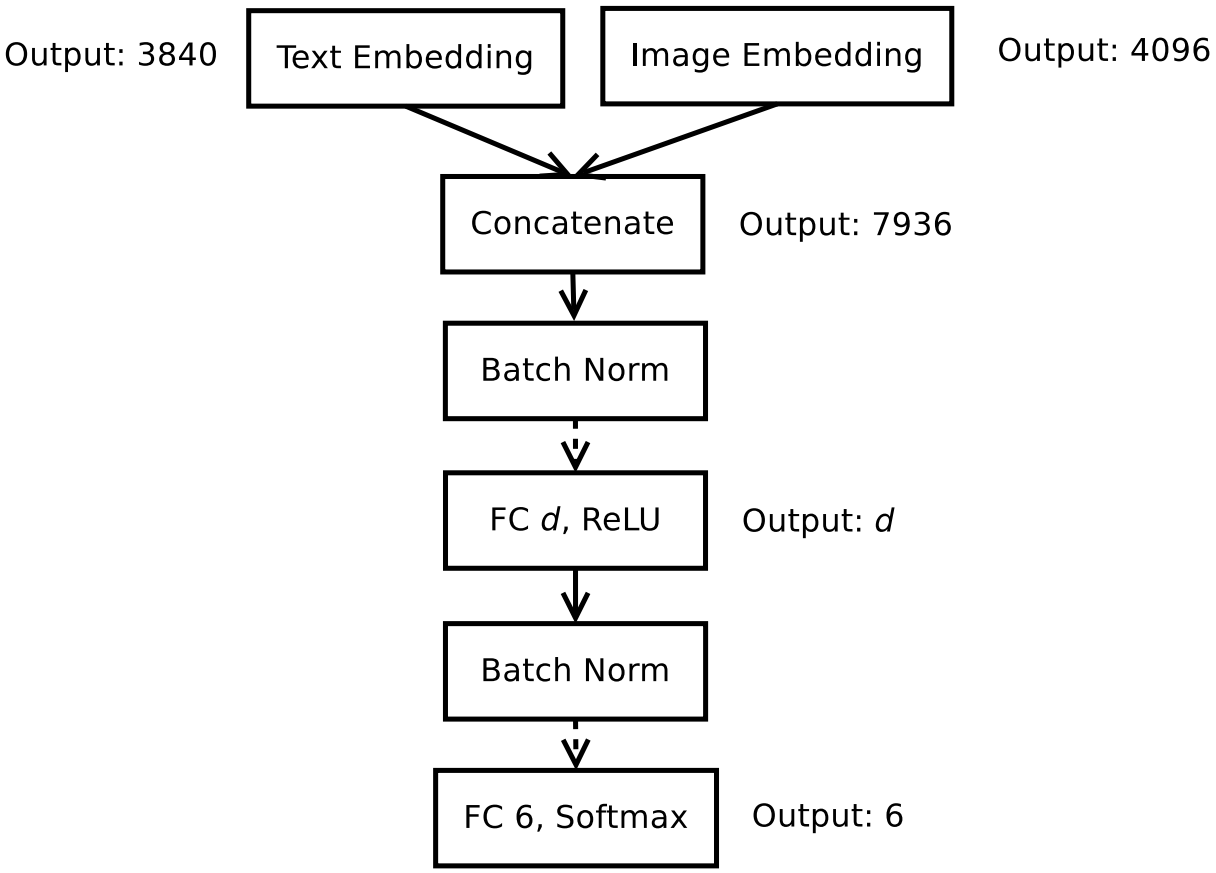


Figure 4.13: Fusion Module (FM): the proposed method for early fusion of textual and visual information. Dashed lines indicate dropout was applied. The hyperparameter d is the number of units in the first fully connected layer.

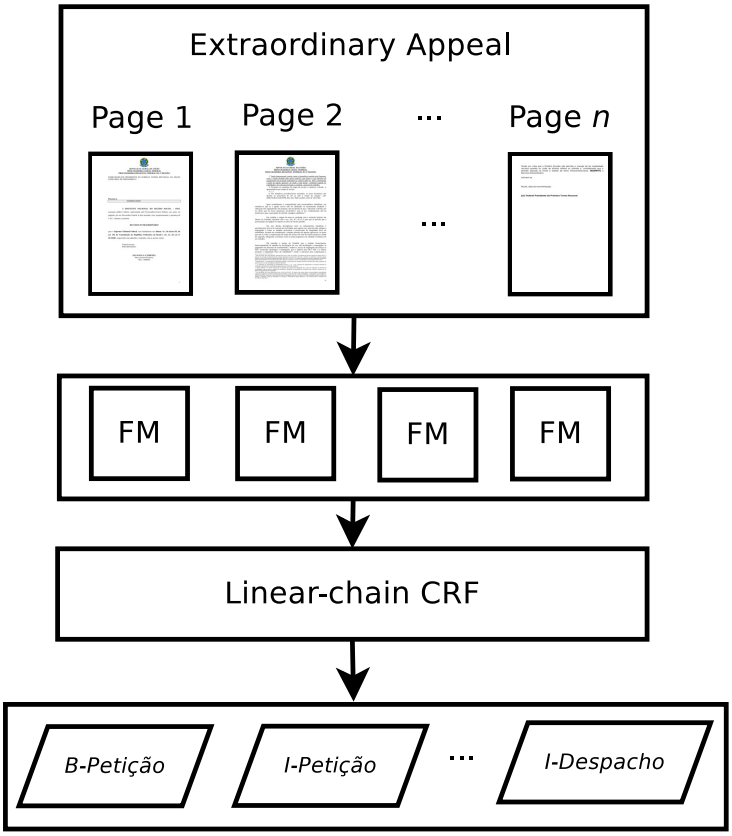


Figure 4.14: Baseline sequence classification method (FM+CRF). We feed the (pre-computed) predictions of the Fusion Module (FM) to a linear-chain CRF to jointly predict the class of each page in an Extraordinary Appeal.

predictions, given a sequence of input pages. We employ the IOB tagging scheme [111] to better leverage the sequential information: we prepend "B-" to the ground truth of first-page samples and "I-" otherwise. For example, if a suit begins with a RE of three pages followed by an ARE of equal length, the label sequence would start with B-RE, I-RE, I-RE, B-ARE, I-ARE, I-ARE.

CRF postprocessing As a baseline method for sequence classification, we save the predictions of the FM for all samples in our data. Then we use these six-dimensional vectors as features to train a linear-chain conditional random field (CRF) [75]. This is similar to what we did in §4.1.4, but using the FM predictions instead of the ones from the CNN. Figure 4.14 illustrates the method (FM+CRF).

BiLSTM As an alternative method for sequence classification of pages, we use a bidirectional long short-term (biLSTM) [51] layer to capture sequential dependencies at the feature extraction level—as opposed to the FM+CRF baseline, which only does so at the prediction level. We experiment with two different kinds of input: the activations of

the first FC layer of the FM (128-dimensional vectors); and the concatenation of the pre-computed image and text embeddings (7,936-dimensional vectors), obtained as described in Section 4.3.4.

The network consists of a biLSTM layer with 128 units for each direction followed by batch normalisation, dropout and an FC layer. When using concatenated image and text embeddings as input, we first apply batch normalisation and dropout, followed by an FC layer with 512 units.

We train four model variants:

1. BiLSTM, which uses fusion activations as input;
2. BiLSTM-CRF, with the same input and a CRF head on top of the described network;
3. BiLSTM-F, which uses concatenated image and text embeddings as input; and
4. BiLSTM-F-CRF, with the same input and a CRF head.

Due to the memory footprint of BiLSTM-F and BiLSTM-F-CRF, we use mini-batches of eight lawsuits when training them. All models are trained for one cycle [132] of 20 epochs. We use the range test to choose learning rates.

4.3.5 Results and discussion

Table 4.12 exhibits the F_1 scores of the best performing models, categorised by whether they use textual (CNN), visual (ResNet50-w and ResNet50), textual and visual (FM), or sequential (FM+CRF and BiLSTM-F) information.

Table 4.12: Test set F_1 scores (in %) of our main approaches for image, text, fusion and sequence classification. Image results are reported for the image test set; all the others, for the text test set.

Class	Majority Baseline		Text	Image		Fusion	Sequence	
	Text	Image	CNN	ResNet50-w	ResNet50	FM	FM+CRF	BiLSTM-F
Acórdão	00.00	00.00	89.96	18.45	06.78	90.74	**91.56**	88.97
ARE	00.00	00.00	55.72	11.33	00.00	57.92	60.74	**61.16**
Despacho	00.00	00.00	62.94	08.44	00.00	63.98	62.69	**64.07**
Others	94.41	94.28	97.31	61.72	95.02	97.24	**97.67**	97.46
RE	00.00	00.00	75.59	32.59	34.96	75.47	78.43	**79.67**
Sentença	00.00	00.00	80.53	43.52	48.67	82.04	83.42	**85.26**
Average	15.73	15.71	77.01	29.34	30.91	77.90	79.09	**79.43**
Weighted	84.41	84.07	94.72	58.09	87.67	94.72	**95.38**	95.30

All models beat majority class classifiers considering both weighted and average F_1 scores—except for ResNet50-w, whose weighted F_1 score is 25.98 p.p. lower. The models

Table 4.13: Text classification: comparison between average and weighted by class frequencies validation set F_1 scores (in %) of the different approaches. The suffix *-w* indicates the use of class frequency penalty weights.

Method	Average F_1	Weighted F_1
CNN-w	64.24	90.98
CNN	**77.14**	**94.37**

with textual data performed much better than those with only visual information available, which is not surprising given that text content is more discriminative than visual aspects when considering the dataset documents—most of them are similar white pages with blocks of text (Figure 4.11).

Regarding fusion and sequence classification results, it is clear that each additional information source contributed to classification metrics: considering average F_1 scores, the FM surpassed the CNN by 0.89 p.p., while the FM+CRF and BiLSTM-F beat the FM by 1.19 and 1.53 p.p., respectively.

In the paragraphs below we will further examine the results of each category (text, image, fusion and sequence) and perform an ablation analysis of the Fusion Module.

Text Classification Results

Table 4.13 compares the validation performance of our approaches for text classification. Using class frequency penalty weights to help with data imbalance did not help: the CNN-w average and weighted scores were 12.90 and 3.39 p.p. lower than its counterpart with no such strategy. Despite using the same architecture, we achieved better results than the ones we reported in §4.1.4. This is probably because we save model parameters only on validation metric improvement when training.

Image Classification Results

Table 4.14 compares the validation performance of the image classification models.

The ResNet50 achieved higher average and weighted scores than its counterpart that uses class frequency penalty weights. That said, since the ResNet50 scores for *Acórdão*, ARE and *Despacho* were zero or close to zero, the ResNet50-w scores were more equally distributed across the different classes. With the intuition that this could lead to more discriminative features, we experiment with both models' activations when fusing textual and visual data.

Table 4.14: Image classification: comparison between validation set F_1 scores (in %) of the different approaches. The suffix *-w* indicates the use of class frequency penalty weights.

Class	ResNet50-w	ResNet50
Acórdão	**17.68**	02.50
ARE	**11.56**	00.00
Despacho	**07.53**	00.00
Others	63.01	**94.77**
RE	32.48	**33.03**
Sentença	43.46	**49.20**
Average	29.29	**29.92**
Weighted	59.13	**87.09**

Table 4.15: Fusion Module: impact of number of hidden units and learnable embeddings for missing data on average validation set F_1 scores (in %). The suffix *-zero* indicates the use of vector of zeros for missing data (as opposed to using learnable embeddings).

Method	Average F_1
FM-512	74.49
FM-512-zero	68.02
FM-128	**75.70**
FM-128-zero	72.95

Image and Text Combination Results

Table 4.15 shows the performance of the FM trained for 10 epochs with different hyper-parameter configurations. Using learnable embeddings for missing textual or visual data proved to be fundamental, improving average F_1 scores by 6.47 and 2.75 p.p. for the models with 512 and 128 hidden units, respectively. While the smaller model performed best, we hypothesise that with further parameter tuning and longer training the bigger model would surpass it.

Table 4.16 compares the scores of the alternative fusion approaches with the ones from the FM. All of them performed much worse, with decreases in average F_1 score ranging from 2.16 to 14.52 p.p. These results signal how the increase in performance seen by the FM is due to the fusion of data sources; not to different training conditions or model capacity—combining visual and textual data helps.

Sequence Classification

Table 4.17 compares the validation performance of the LSTM models. To ensure a fair comparison to the other approaches, though we use IOB tagging scheme during training, when reporting results we consider only the original classes. If a given page is an ARE, for

Table 4.16: Fusion Module ablation, comparing the test set F_1 scores (in %) of the hybrid classifier and of a version of the fusion module that ignores image activations (w/o img acts), that is, always uses the missing image embedding. For the hybrid classifier, we report results using both image classifiers: with (HC-w) and without (HC) class frequency penalty. Between parentheses, the difference in performance compared with using the original fusion module (FM).

Class	Text test split		Text + image test split		
	FM	fusion w/o img acts	FM	HC-w	HC
Acórdão	90.74	88.27 (-2.47)	88.5	41.36 (-47.14)	87.68 (-0.82)
ARE	57.92	54.09 (-3.83)	56.6	49.02 (-7.58)	43.91 (-12.69)
Despacho	63.98	62.01 (-1.97)	63.79	42.71 (-21.08)	61.85 (-1.94)
Others	97.24	97.27 (+0.03)	97.03	95.80 (-1.23)	97.02 (-0.01)
RE	75.47	73.26 (-2.21)	75.05	72.11 (-2.94)	75.00 (-0.05)
Sentença	82.04	79.58 (-2.46)	81.21	74.07 (-7.14)	79.68 (-1.53)
Average	77.90	75.74 (-2.16)	77.03	62.51 (-14.52)	74.19 (-2.84)
Weighted	94.72	94.47 (-0.25)	94.32	92.58 (-1.74)	93.95 (-0.37)

example, the predictions B-ARE and I-ARE would both be considered correct, regardless of the position of the page in its lawsuit.

Table 4.17: Sequence classification: comparison between average and weighted by class frequencies validation set F_1 scores (in %) of the different approaches.

Method	Average F_1	Weighted F_1
BiLSTM	77.16	94.25
BiLSTM-CRF	78.45	94.46
BiLSTM-F	**79.03**	**94.81**
BiLSTM-F-CRF	78.87	94.58

The variants that use as input the image and text embeddings (BiLSTM-F and BiLSTM-F-CRF) outperformed the ones that use the FM activations (BiLSTM and BiLSTM-CRF). This suggests that it is beneficial to jointly learn how to consider sequential dependencies and how to combine multi-modal information. Surprisingly, the CRF layer helped the BiLSTM model, with an increase in 1.29/0.25 average/weighted F_1 scores, but not the BiLSTM-F model. This may be an artifact of our training settings, with its limited number of training epochs.

First page evaluation

Table 4.18 shows the difference in classification performance of samples that are the first page of a document versus those that are not, considering all levels of data availability (text, image, and fusion).

Table 4.18: Comparison of first page and not first page of a document classification performance. We report test set F_1 scores (in %) for image, text, and fusion classification using as models the CNN, the ResNet50-w and the FM, respectively. Between parentheses, the number of samples.

Class	Text		Image		Fusion	
	First page	Not first page	First page	Not first page	First page	Not first page
Acórdão	**92.47 (199)**	83.66 (74)	**34.28 (197)**	08.24 (88)	**93.40 (199)**	77.19 (88)
ARE	47.65 (213)	**56.74 (1,628)**	06.71 (203)	**12.10 (2,334)**	**59.95 (213)**	56.28 (2,442)
Despacho	**71.54 (147)**	40.43 (51)	**12.59 (146)**	03.68 (52)	**71.81 (147)**	40.45 (52)
Others	**99.02 (25,744)**	96.58 (59,664)	**78.19 (24,193)**	54.29 (63,709)	**99.04 (25,744)**	96.26 (66,789)
RE	74.45 (312)	**75.65 (6,019)**	18.28 (301)	**33.72 (5,876)**	**75.50 (312)**	75.03 (6,074)
Sentença	**81.47 (265)**	80.32 (1,210)	26.61 (262)	**49.71 (1,216)**	**83.11 (265)**	80.78 (1,238)
Average	**77.77 (26,880)**	72.23 (68,646)	**29.44 (25,302)**	26.96 (73,275)	**80.47 (26,880)**	71.00 (76,683)
Weighted	**97.96 (26,880)**	93.46 (68,646)	**75.65 (25,302)**	51.13 (73,275)	**98.11 (26,880)**	93.00 (76,683)

The first page sample set obtained average/weighted F_1 scores 5.54/4.50, 2.48/24.52 and 9.47/5.11 p.p. higher than its complement, for the text, image and fusion levels, respectively. These results confirm our hypothesis that the first pages are more informative from the point of view of both textual and visual data. Therefore, one possible improvement for page classification of the legal documents is training under a multi-task setting that jointly learns to classify pages and establish document boundaries.

4.3.6 Summary

In this section, we presented a novel dataset of Brazilian lawsuits with visual and textual data and proposed a method for sequence-aware multimodal classification of pages from legal documents. Our proposed Fusion Module combines visual and textual features extracted from convolutional neural networks trained separately on image and text data. We experiment with two approaches for sequence classification: post-processing the predictions of the Fusion Module using a linear-chain conditional random field; and training bidirectional LSTM models that alternatively use as input Fusion Module activations or the concatenation of image and text embeddings. Our Fusion Module outperformed the unimodal models, with an ablation analysis confirming that improvement is due to the combination of modalities. We find that learning embeddings for missing visual or textual input is much better than using a vector of zeroes for such cases. Our experiments also confirm the intuition that the first page of a document is easier to classify. Sequence classification of pages brought further improvements, with the best performing model jointly learning how to combine modalities and consider sequential dependencies.

Therefore, future work would include the end-to-end training of the full pipeline: image and text feature extractors, Fusion Module and sequence modelling. Moreover, it is

worthwhile to explore if transformer-based [143] text encoders such as BERT [31] and T5 [110] can further improve classification performance.

4.4 Conclusions

In this chapter we have proposed a dataset of Supreme Court documents. 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 fusing visual, textual and sequential information lead to better performing models.

Our topic modelling semantic analysis indicates that the detected topics captures aspects of legal matters. On the other hand, we observed a trade-off: introducing more topics can be useful yields topics with finer semantics; however, a greater number of topics may increase the likelihood of meaningless (OCR artifacts) or jumbled (miscellanea) topics. Regarding the quantitative analysis, the topic representation yielded reasonable results—further evidence of their usefulness for case management.

In the next chapter, we will examine a dataset of official gazette documents with labelled and unlabelled texts.

Chapter 5

DODF dataset

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) [57] 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 training and inference time and being less computationally expensive. Finally, we conduct ablation analysis to assess the performance impact of the ULMFiT parts[1].

5.1 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.

[1] An early version of this section has been published in: Pedro H. Luz de Araujo, Teófilo E. de Campos, and Marcelo Magalhães Silva de Sousa: Inferring the source of official texts: can SVM beat ULMFiT? [89].

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, 19]. Other works [66, 37, 74, 68] explore the use of automatic summarisation to mitigate the amount of information legal professional have to process. Text classification has been utilised for decision prediction [3, 67], area of legal practice attribution [135] and fine-grained legal-issue classification. Some effort has been applied to the processing of Brazilian legal documents [26, 28, 87], as we previously discussed (Chapter 4).

In this chapter, we aim to identify which public entity originated documents fom the Official Gazette of the Federal District, a document classification task, as defined in Chapter 4. Our objectives are:

- to construct a dataset of documents from the Official Gazette of the Federal District from Brazil with labels for public body of origin;
- to train bag-of-words and ULMFiT [57] models on the data and compare the results;
- to perform a ablation analysis of the ULMFiT model.

Our hypotheses are twofold:

1. bag-of-words and ULMFiT models will outperform a majority class classifier;
2. ULMFiT will outperform the bag-of-words approaches.

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 [57] 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) [31] and Generative Pre-trained Transformer 2 (GPT-2) [109]. Our main contributions[2] are:

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

[2]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.

5.2 The DODF dataset

The DODF[3] data consists of 2,652 texts extracted from editions of the Official Gazette of the Federal District[4] published online in January 2019 in PDF form. 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 5.1 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 5.4.

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.

[3] *Diário Oficial do DF*—Official Gazette of the Federal District.
[4] Published at `https://www.dodf.df.gov.br/`.

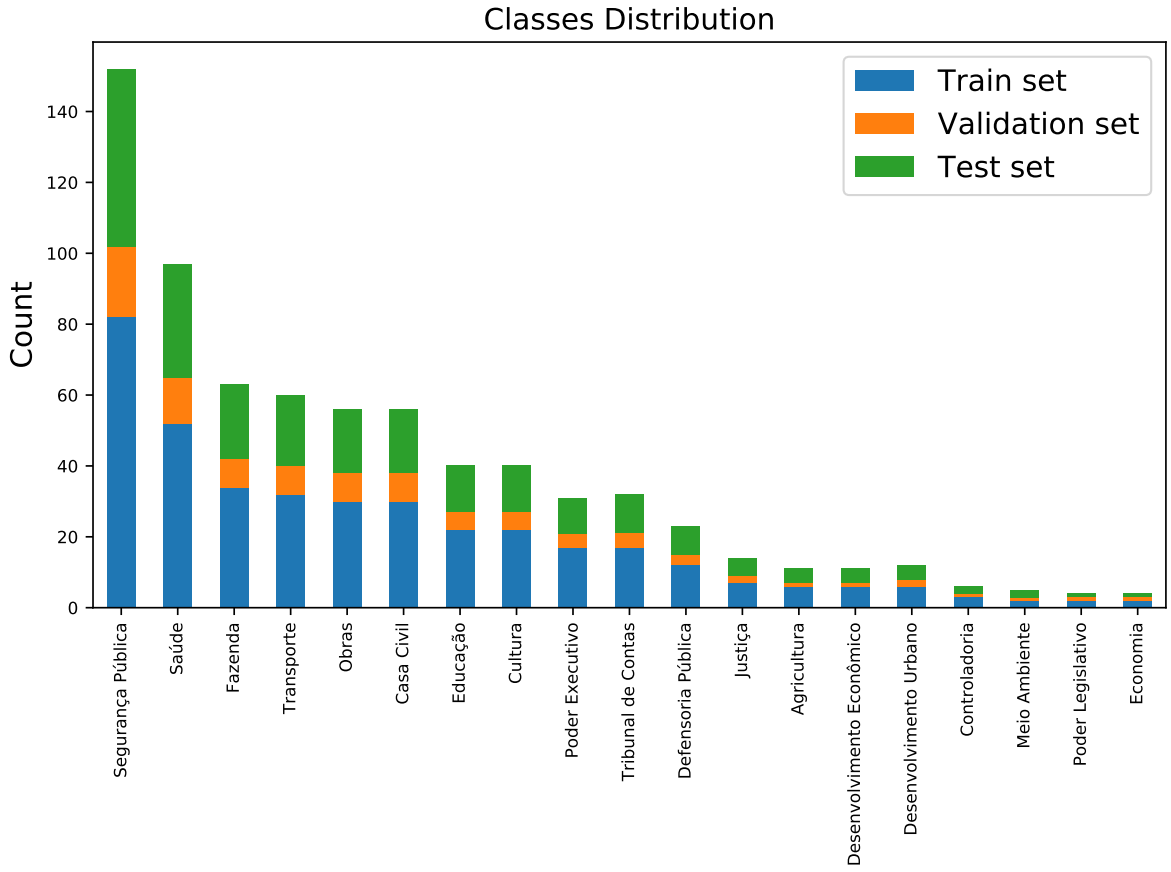


Figure 5.1: Class counts for each DODF data split.

5.3 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.

5.3.1 Preprocessing

We first lowercase the text and use SentencePiece [73] to tokenize it. We chose SentencePiece because that was the tokenizer used for the pre-trained language model (more about that on Section 5.3.3), 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.

5.3.2 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 through equation 2.66. All document vectors are normalised to have unit Euclidean norm.

We use the obtained BOW to train these shallow classifiers: SVM [49] with linear kernel and NB[5].

5.3.3 Transfer learning

We use ULMFiT [57] to leverage information contained in the unlabelled language model dataset[6]. 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:

[5]The baseline experiments are implemented using the scikit-learn library [17].
[6]The transfer learning experiments are implementend using PyTorch [104] and FastAI [56].

Language model pre-training We use a bidirectional Portuguese language model[7] 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) [13] 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 [57], 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 [132] 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 [60], dropout [134] and a fully connected layer. The first fully connected layer has 50 units and ReLU [99] 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:

$$\mathbf{h}_c = \text{concat}(\mathbf{h}_t, \text{maxpool}(\mathbf{H}), \text{averagepool}(\mathbf{H}))\,. \tag{5.1}$$

5.4 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.

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

5.4.1 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 (Eq. 4.1) to compensate class imbalance and giving all classes the same weight.

5.4.2 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 [131], 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 [69] 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 [83] of 0.1, backpropagation through time of length 70 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 [57].

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 [57] 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

Table 5.1: 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

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 (Eq. 4.1). Nevertheless, this did not contribute to significant changes in classification metrics.

5.5 Results

Table 5.1 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 (after hyperparameter tuning), the ULMFiT model took more than half an hour—not counting the language model pre-training, which took hours[8]. 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.

5.5.1 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 5.2 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.

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,

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

Table 5.2: 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)
One-step transfer (f)	11.46 (-72.00)	24.59 (-64.31)	34.18 (-54.85)
One-step transfer (b)	12.29 (-68.31)	27.35 (-61.53)	38.40 (-51.05)
One-step transfer (f+b)	12.36 (-71.38)	26.35 (-64.53)	37.97 (-53.59)

since gradual freezing was shown to be beneficial in the paper that proposed ULMFiT [57]. 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.

One-step 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 hypothesise 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.

5.6 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 when considering both F_1 and accuracy scores. 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.

5.7 Conclusions

In this chapter we have proposed a text classification dataset of documents from the Official Gazette of the Federal District. We have found that in the task presented, where word order is not of utmost importance, the SVM is still competitive when compared to deep, state-of-the-art models that leverage unsupervised pre-training.

Chapter 6

Conclusions

One of the major challenges NLP research faces is model generalisation. It is not uncommon that models that perform surprisingly well in their training domain fall apart when applied to other domains. Research efforts that aim to shorten this gap—studies on domain adaption and transfer learning, for example—require data from different domains for training and evaluating new methods.

In this work we have presented three novel domain-specific datasets in Brazilian Portuguese. We have focused on documents from the legal and public administration domain and on text classification and named entity recognition tasks. For each dataset, we have trained and evaluated models ranging from traditional bag-of-words representation to deep neural networks. We hope that our results can serve as a basis of comparison for future work on the data.

Although the main contributions of this work are the datasets, our experiments have resulted in the following findings, which we list as empirical contributions:

- A BiLSTM-CRF model trained on the LeNER-Br data is able to recognise domain-specific entities as well as general entities with no need for any specific pre-processing and feature engineering (§3).
- BOW models can achieve classification performance on par with deep learning models, specially in scenarios with less data such as the Small VICTOR (§4.1) and DODF (§5) ones.
- Topics detected using LDA can be used as a starting point to help with STF's case management (§4.2).
- VICTOR's document type classification performance improves with each additional input modality (visual features and sequential information) (§4.3).

The models we trained and evaluated were intended to serve only as benchmarks to support and encourage future work. Given that and our limited computational resources, we were not able to perform extensive hyperparameter tuning for the deep neural network models. This may explain the surprisingly limited benefit of using SOTA methods such as ULMFiT when compared with simple BOW models trained with an SVM classifier. A possible future direction is to run additional experiments with more varied hyperparameter configurations to try to achieve better results. Moreover, comparing our results with ones obtained using more recent techniques such as transformer-based models and pretrained contextual embeddings would be another worthwhile pursuit.

Since previous attempts at combining VICTOR's image and text data faced optimisation difficulties when jointly training the full pipeline, we opted to initially train each of our modules separately. Now that there is a well-working model as a proof-of-concept and given that the best performing variant jointly learned fusion and sequence processing, it may be fruitful to further explore end-to-end training all modules (text and image feature extractors, and fusion and sequential modules).

Due to time and human resources constraints, documents were not annotated by more than one person, which precluded computing metrics for annotator agreement. In the absence of such objective measurements, annotation consistency and correctness was enforced by thorough revising in the case of the LeNER-Br and Official Gazette datasets. VICTOR's labelling was executed during the ordinary workflow of the Court staff, so we are not aware of the labelling procedure details. But given the cost of failure, we believe those are highly accurate labels.

Finally, we hope that our data encourages future research on transfer learning, domain adaptation and generalisation, and cross-lingual learning for texts in the Portuguese language.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 33

[2] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media* (USA, 2011), Association for Computational Linguistics, p. 30–38. 43

[3] Aletras, N., Tsarapatsanis, D., Preotiuc-Pietro, D., and Lampos, V. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ in Computer Science* (10 2016). 44, 62, 86

[4] Audebert, N., Herold, C., Slimani, K., and Vidal, C. Multimodal deep networks for text and image-based document classification. *CoRR abs/1907.06370* (2019). 72

[5] Augenstein, I., Derczynski, L., and Bontcheva, K. Generalisation in named entity recognition. *Computer Speech and Language 44*, C (July 2017), 61–83. 30

[6] Bird, S., Klein, E., and Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009. 32

[7] Blei, D. M. Probabilistic topic models. *Commun. ACM 55*, 4 (Apr. 2012), 77–84. 61, 62

[8] Blei, D. M., and Lafferty, J. D. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning* (New York, NY, USA, 2006), ICML, ACM, pp. 113–120. 62

[9] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3* (Mar. 2003), 993–1022. 62, 63

[10] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* (2016). 1, 72, 121

[11] Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation* (Copenhagen, Denmark, Sept. 2017), Association for Computational Linguistics, pp. 169–214. 1

[12] Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (New York, NY, USA, 1992), COLT, Association for Computing Machinery, p. 144–152. 13

[13] Bradbury, J., Merity, S., Xiong, C., and Socher, R. Quasi-recurrent neural networks. *CoRR abs/1611.01576* (2016). 90

[14] Braz, F. A., da Silva, N. C., de Campos, T. E., Chaves, F. B. S., Ferreira, M. H. S., Inazawa, P. H., Coelho, V. H. D., Sukiennik, B. P., de Almeida, A. P. G. S., Vidal, F. B., Bezerra, D. A., Gusmao, D. B., Ziegler, G. G., Fernandes, R. V. C., Zumblick, R., and Peixoto, F. H. 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. 45

[15] Breiman, L. Random forests. *Mach. Learn. 45*, 1 (Oct. 2001), 5–32. 14

[16] Broscheit, S. 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)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 677–685. 114, 115, 116, 120

[17] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122. 49, 89

[18] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122. 49

[19] Cardellino, C., Teruel, M., Alonso Alemany, L., and Villata, S. 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`. 31, 44, 62, 86

[20] Carter, D. J., Brown, J., and Rahmani, A. 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* (2016), 1300. 44, 63

[21] Chen, N., and Blostein, D. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal of Document Analysis and Recognition (IJDAR) 10*, 1 (Jun 2007), 1–16. 72

[22] Chen, T., and Guestrin, C. XGBoost: A scalable tree boosting system. *CoRR abs/1603.02754* (2016). 14, 56, 64

[23] Chollet, F., et al. Keras. `https://keras.io`, 2015. 50, 74

[24] Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. 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* (Valencia, Spain, Apr. 2017), Association for Computational Linguistics, pp. 1107–1116. 44, 50

[25] Cortes, C., and Vapnik, V. Support-vector networks. *Mach. Learn. 20*, 3 (Sept. 1995), 273–297. 13

[26] da Silva, N. C., Braz, F. A., de Campos, T. E., Gusmao, D., Chaves, F., Mendes, D., Bezerra, D., Ziegler, G., Horinouchi, L., Ferreira, M., Carvalho, G., Fernandes, R. V. C., Peixoto, F. H., Filho, M. S. M., Sukiennik, B. P., Rosa, L. S., Silva, R. Z. M., and Junquilho, T. A. 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. 45, 61, 63, 86

[27] de Castilho, R. E., Mujdricza-Maydt, E., Yimam, S. M., Hartmann, S., Gurevych, I., Frank, A., and Biemann, C. 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)* (2016), pp. 76–84. 32

[28] de Vargas Feijó, D., and Moreira, V. P. Rulingbr: A summarization dataset for legal texts. In *Computational Processing of the Portuguese Language* (Cham, 2018), A. Villavicencio, V. Moreira, A. Abad, H. Caseli, P. Gamallo, C. Ramisch, H. Gonçalo Oliveira, and G. H. Paetzold, Eds., Springer International Publishing, pp. 255–264. 61, 63, 86

[29] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. *Journal of The American Society for Information Science 41*, 6 (1990), 391–407. 62, 72

[30] Deshpande, V. P., Erbacher, R. F., and Harris, C. An evaluation of naïve bayesian anti-spam filtering techniques. In *IEEE SMC Information Assurance and Security Workshop* (June 2007), pp. 333–340. 43

[31] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805* (2018). 1, 84, 86, 115, 119, 121

[32] Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., and Wudali, R. Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*. Springer, 2010, pp. 27–43. 31, 44, 62, 86

[33] Elman, J. L. Finding structure in time. *Cognitive Science 14*, 2 (1990), 179–211. 22

[34] Engin, D., Emekligil, E., Oral, B., Arslan, S., and Akpinar, M. Multimodal deep neural networks for banking document classification. In *International Conference on Advances in Information Mining and Management* (2019), pp. 21–25. 72, 73

[35] Eshel, Y., Cohen, N., Radinsky, K., Markovitch, S., Yamada, I., and Levy, O. Named entity disambiguation for noisy text. In *Proceedings of the 21st Conference on Computational Natural Language Learning* (Vancouver, Canada, Aug. 2017), Association for Computational Linguistics, pp. 58–68. 122

[36] Freitas, C., Mota, C., Santos, D., Oliveira, H. G., and Carvalho, P. Second HAREM: Advancing the state of the art of named entity recognition in Portuguese. In *Language Resources and Evaluation Conference (LREC)* (2010), European Language Resources Association. 30

[37] Galgani, F., Compton, P., and Hoffmann, A. Combining different summarization techniques for legal text. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data* (Stroudsburg, PA, USA, 2012), HYBRID, Association for Computational Linguistics, pp. 115–123. 44, 62, 86

[38] Ganea, O.-E., and Hofmann, T. Deep joint entity disambiguation with local neural attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Copenhagen, Denmark, Sept. 2017), Association for Computational Linguistics, pp. 2619–2629. 114, 115, 116, 117, 119, 120, 121

[39] Genthial, G. Sequence tagging - named entity recognition with Tensorflow. GitHub repository `https://github.com/guillaumegenthial/sequence_tagging/tree/0048d604f7a4e15037875593b331e1268ad6e887`, 2017. 33

[40] Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: continual prediction with lstm. In *International Conference on Artificial Neural Networks* (1999), vol. 2, pp. 850–855 vol.2. 23

[41] Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldridge, J., Ie, E., and Garcia-Olano, D. Learning dense representations for entity retrieval, 2019. 114, 118, 119

[42] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 5, 13, 23

[43] Graves, A., and Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks 18*, 5-6 (2005), 602–610. 33, 50

[44] Grimmer, J., and Stewart, B. M. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis 21*, 3 (2013), 267–297. 65

[45] Hajek, B. *Random Processes for Engineers*. Cambridge University Press, 2015. 5

[46] Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., and Aluisio, S. 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. 27, 33

[47] Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference and prediction*, 2 ed. Springer, 2009. 56

[48] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778. 75

[49] Hearst, M. A. Support vector machines. *IEEE Intelligent Systems 13*, 4 (July 1998), 18–28. 89

[50] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780. 23

[51] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780. 33, 50, 78

[52] Hoekstra, R., Breuker, J., Bello, M. D., and Boer, A. The LKIF Core ontology of basic legal concepts. In *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques* (2007). 32

[53] Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Edinburgh, Scotland, UK., July 2011), Association for Computational Linguistics, pp. 782–792. 117, 122

[54] Hoffman, M. D., Blei, D. M., and Bach, F. Online Learning for Latent Dirichlet Allocation. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1* (USA, 2010), NIPS, Curran Associates Inc., pp. 856–864. 64

[55] Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 1999), SIGIR, ACM, pp. 50–57. 62

[56] Howard, J., et al. fastai. `https://github.com/fastai/fastai`, 2018. 74, 89

[57] Howard, J., and Ruder, S. 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)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 328–339. 1, 75, 85, 86, 89, 90, 91, 94

[58] Howard, J., and Ruder, S. 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)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 328–339. 44

[59] Huang, Z., Xu, W., and Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *CoRR abs/1508.01991* (2015). 53

[60] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37* (2015), JMLR.org, pp. 448–456. 76, 90

[61] Jain, R., and Wigington, C. Multimodal document image classification. In *International Conference on Document Analysis and Recognition (ICDAR)* (2019), pp. 71–77. 72, 73

[62] Ji, H., Grishman, R., Dang, H. T., Griffitt, K., and Ellis, J. Overview of the TAC 2010 knowledge base population track. In *Text Analysis Conference* (2010), vol. 3. 122

[63] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML* (Berlin, Heidelberg, 1998), C. Nédellec and C. Rouveirol, Eds., Springer Berlin Heidelberg, pp. 137–142. 43

[64] Johnson, R., and Zhang, T. 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* (2016), ICML, JMLR.org, pp. 526–534. 44

[65] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. 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* (2017), Association for Computational Linguistics, pp. 427–431. 43

[66] Kanapala, A., Pal, S., and Pamula, R. Text summarization from legal documents: a survey. *Artificial Intelligence Review* (Jun 2017). 44, 62, 86

[67] Katz, D. M., Bommarito, Michael J, I., and Blackman, J. Predicting the Behavior of the Supreme Court of the United States: A General Approach. *arXiv e-prints* (Jul 2014), arXiv:1407.6333. 44, 62, 86

[68] Kim, M.-Y., Xu, Y., and Goebel, R. Summarization of legal texts with high cohesion and automatic compression rate. In *New frontiers in artificial intelligence* (2013), Springer. 44, 62, 86

[69] Kingma, D. P., and Ba, J. Adam: A method for stochastic optmisation. In *International Conference on Learning Representations (ICLR)* (2015). 20, 91

[70] Kingma, D. P., and Ba, J. Adam: A method for stochastic optmisation. In *International Conference on Learning Representations (ICLR)* (2015). Preprint available at `https://arxiv.org/abs/1412.6980`. 35, 50, 74

[71] Kolitsas, N., Ganea, O.-E., and Hofmann, T. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning* (Brussels, Belgium, Oct. 2018), Association for Computational Linguistics, pp. 519–529. 113, 114, 115, 117, 120

[72] Korobov, M. Keras. `https://sklearn-crfsuite.readthedocs.io/en/latest/`, 2015. 53

[73] Kudo, T., and Richardson, J. 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)* (Brussels, Belgium, Nov. 2018), Association for Computational Linguistics (ACL), pp. 66–71. 25, 89

[74] Kumar, R., and Raghuveer, K. Legal document summarization using latent dirichlet allocation. *International Journal of Computer Science and Telecommunications 3* (2012), 114–117. 44, 62, 86

[75] Lafferty, J. D., andrew, M., and Pereira, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning* (San Francisco, CA, USA, 2001), ICML, Morgan Kaufmann Publishers Inc., pp. 282–289. 14, 15, 33, 53, 78

[76] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. 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* (San Diego, California, June 2016), Association for Computational Linguistics, pp. 260–270. 30, 33, 35, 36, 53

[77] Le, P., and Titov, I. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 1935–1945. 114, 116, 117, 118, 120

[78] Le, P., and Titov, I. Distant learning for entity linking with automatic noise detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 4081–4090. 114, 118

[79] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (1989), 541–551. 23

[80] Liu, C., Hsaio, W., Lee, C., Chang, T., and Kuo, T. Semi-supervised text classification with universum learning. *IEEE Transactions on Cybernetics 46*, 2 (Feb 2016), 462–473. 44

[81] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR abs/1907.11692* (2019). 1

[82] Logeswaran, L., Chang, M.-W., Lee, K., Toutanova, K., Devlin, J., and Lee, H. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 3449–3460. 113, 114, 115, 118, 119, 121, 122

[83] Loshchilov, I., and Hutter, F. Fixing weight decay regularization in Adam. *CoRR abs/1711.05101* (2017). 91

[84] Luo, G., Huang, X., Lin, C.-Y., and Nie, Z. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), Association for Computational Linguistics, pp. 879–888. 115

[85] Luz de Araujo, P. H., and de Campos, T. E. Topic modelling brazilian supreme court lawsuits. In *International Conference on Legal Knowledge and Information Systems (JURIX)* (Prague, Czech Republic, December 9-11 2020), Frontiers in Artificial Intelligence and Applications, IOS Press, pp. 113–122. viii, 3, 60

[86] Luz de Araujo, P. H., de Campos, T. E., Ataides Braz, F., and Correia da Silva, N. VICTOR: a dataset for Brazilian legal documents classification. In *Proceedings of The 12th Language Resources and Evaluation Conference* (Marseille, France, May 2020), European Language Resources Association, pp. 1449–1458. viii, 3, 41, 72, 73

[87] Luz de Araujo, P. H., de Campos, T. E., de Oliveira, R. R. R., Stauffer, M., Couto, S., and Bermejo, P. 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). vii, 3, 29, 44, 61, 62, 63, 86

[88] Luz de Araujo, P. H., de Campos, T. E., and Magalhaes Silva de Sousa, M. Inferring the source official texts: can SVM beat ULMFiT? In *International Conference on the Computational Processing of Portuguese (PROPOR)* (Evora, Portugal, March 2-4 2020), Lecture Notes on Computer Science (LNCS), Springer. Code and data available from `https://cic.unb.br/~teodecampos/KnEDLe/`. viii, 3

[89] Luz de Araujo, P. H., de Campos, T. E., and Magalhaes Silva de Sousa, M. Inferring the source official texts: can SVM beat ULMFiT? In *International Conference on the Computational Processing of Portuguese (PROPOR)* (Evora, Portugal, March 2-4 2020), Lecture Notes on Computer Science (LNCS), Springer. Code and data available from `https://cic.unb.br/~teodecampos/KnEDLe/`. 85

[90] Ma, X., and Hovy, E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Berlin, Germany, August 7-12 2016), ACL, pp. 1064–1074. Preprint available at `https://arxiv.org/abs/1603.01354`. 30

[91] MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002. 5

[92] Mansouri, A., Affendey, L. S., and Mamat, A. Named entity recognition approaches. *International Journal of Computer Science and Network Security 8*, 2 (2008), 339–344. 30

[93] Mauá, D. D. Modelos de tópicos na classificação automática de resenhas de usuários. Master's thesis, Escola Politécnica da Universidade de São Paulo, 2009. 61

[94] Mendonça Jr., C. A. E., Macedo, H., Bispo, T., Santos, F., Silva, N., and Barbosa, L. Paramopama: a Brazilian-Portuguese corpus for named entity recognition. In *XII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)* (2015), SBC. 3, 29, 30, 31, 35

[95] Mendonça Jr., C. A. E. M., Barbosa, L. A., and Macedo, H. T. 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)* (2016), SBC. 35, 36

[96] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119. 1, 121

[97] Mondal, I., Purkayastha, S., Sarkar, S., Goyal, P., Pillai, J., Bhattacharyya, A., and Gattu, M. Medical entity linking using triplet network. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop* (Minneapolis, Minnesota, USA, June 2019), Association for Computational Linguistics, pp. 95–100. 121

[98] Mota, C., Lima, A., Nascimento, A., Miranda, P., and de Mello, R. Classificação de páginas de petições iniciais utilizando redes neurais convolucionais multimodais. In *Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional* (Porto Alegre, RS, Brasil, 2020), SBC, pp. 318–329. 72, 73

[99] Nair, V., and Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICLR)* (USA, 2010), Omnipress, pp. 807–814. 16, 90

[100] Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence 194* (2013), 151–175. 30, 36, 37

[101] Onoe, Y., and Durrett, G. Fine-Grained Entity Typing for Domain Independent Entity Linking. *arXiv e-prints* (Sept. 2019), arXiv:1909.05780. 114, 117, 119, 122

[102] O'Neill, J., Robin, C., O'Brien, L., and Buitelaar, P. 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). 44, 63

[103] Pappu, A., Blanco, R., Mehdad, Y., Stent, A., and Thadani, K. Lightweight multilingual entity extraction and linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2017), WSDM, Association for Computing Machinery, p. 365–374. 114, 115, 117, 119

[104] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. 74, 89

[105] Pennington, J., Socher, R., and Manning, C. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543. 1, 27, 33

[106] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. 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)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 2227–2237. 1

[107] Qian, N. On the momentum term in gradient descent learning algorithms. *Neural networks: the official journal of the International Neural Network Society 12*, 1 (Jan. 1999), 145–151. 20

[108] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. 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

[109] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog 1*, 8 (February 2019). 1, 86

[110] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research 21*, 140 (2020), 1–67. 84

[111] Ramshaw, L. A., and Marcus, M. P. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, 1999, pp. 157–176. Preprint available at http://arxiv.org/abs/cmp-lg/9505040. 32, 53, 78

[112] Ratinov, L., Roth, D., Downey, D., and Anderson, M. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA, June 2011), Association for Computational Linguistics, pp. 1375–1384. 113

[113] Řehůřek, R., and Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC Workshop on New Challenges for NLP Frameworks* (Valletta, Malta, May 2010), ELRA, pp. 45–50. 64

[114] Remmits, Y. Finding the Topics of Case Law: Latent Dirichlet Allocation on Supreme Court Decisions, 2017. Bachelor's thesis, Radboud University, July 2017. 44, 63

[115] Rennie, J. D. M., Shih, L., Teevan, J., and Karger, D. R. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* (2003), ICML'03, AAAI Press, p. 616–623. 12

[116] Rubin, T. N., Chambers, A., Smyth, P., and Steyvers, M. Statistical topic models for multi-label document classification. *Machine Learning 88*, 1 (Jul 2012), 157–208. 61

[117] Ruder, S. An overview of gradient descent optimization algorithms. *CoRR abs/1609.04747* (2016). 20

[118] Ruder, S. *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway, 2019. 1, 22

[119] Ruder, S. *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway, 2019. 18, 28

[120] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning Representations by Back-propagating Errors. *Nature 323*, 6088 (1986), 533–536. 20

[121] Rusiñol, M., Frinken, V., Karatzas, D., Bagdanov, A. D., and Lladós, J. Multimodal page classification in administrative document image streams. *International Journal on Document Analysis and Recognition (IJDAR) 17*, 4 (Dec 2014), 331–341. 72, 73

[122] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*, 3 (2015), 211–252. 75

[123] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 72

[124] Santos, D., and Cardoso, N. A golden resource for named entity recognition in Portuguese. In *International Workshop on Computational Processing of the Portuguese Language* (2006), Springer, pp. 69–79. 30, 36, 37

[125] Schapire, R. E. The strength of weak learnability. *Mach. Learn. 5*, 2 (July 1990), 197–227. 14

[126] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015). 121

[127] Secretaria de Comunicação Social do Conselho Nacional de Justiça. Sumário executivo do relatório justiça em números 2020, 2018. 41

[128] Sievert, C., and Shirley, K. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces* (Baltimore, Maryland, USA, June 2014), Association for Computational Linguistics, pp. 63–70. 65

[129] Sil, A., and Yates, A. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (New York, NY, USA, 2013), CIKM, Association for Computing Machinery, p. 2369–2374. 115

[130] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations* (2015). 72

[131] Smith, L. N. No more pesky learning rate guessing games. *CoRR abs/1506.01186* (2015). 21, 75, 91

[132] Smith, L. N., and Topin, N. Super-convergence: Very fast training of residual networks using large learning rates. *CoRR abs/1708.07120* (2017). 75, 79, 90

[133] Smith, R. An overview of the Tesseract OCR engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR)* (2007), vol. 2, IEEE, pp. 629–633. 45, 73

[134] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res. 15*, 1 (Jan. 2014), 1929–1958. 33, 90

[135] Şulea, O.-M., Zampieri, M., Vela, M., and van Genabith, J. Predicting the law area and decisions of French Supreme Court cases. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP* (2017), INCOMA Ltd., pp. 716–722. 44, 62, 86

[136] Supremo Tribunal Federal. Ministra Cármen Lúcia anuncia início de funcionamento do Projeto Victor, de inteligência artificial, 2018. 41

[137] Tjong Kim Sang, E. F., and De Meulder, F. 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* (2003), vol. 4, Association for Computational Linguistics, pp. 142–147. 32, 33

[138] Tsai, C.-T., and Roth, D. 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* (San Diego, California, June 2016), Association for Computational Linguistics, pp. 589–598. 114, 116, 117

[139] Undavia, S., Meyers, A., and Ortega, J. E. A comparative study of classifying legal documents with neural networks. In *Federated Conference on Computer Science and Information Systems (FedCSIS)* (Sep. 2018), pp. 515–522. 44, 62

[140] Upadhyay, S., Gupta, N., and Roth, D. Joint multilingual supervision for cross-lingual entity linking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 2486–2495. 114, 117, 118

[141] van der Maaten, L., and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605. 118

[142] Van Rossum, G., and Drake, F. L. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. 33

[143] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. 1, 84

[144] Wang, C., Zhang, M., Ma, S., and Ru, L. Automatic online news issue construction in web environment. In *Proceedings of the 17th International Conference on World Wide Web* (New York, NY, USA, 2008), Association for Computing Machinery, p. 457–466. 43

[145] Wiedemann, G., and Heyer, G. Multi-modal page stream segmentation with convolutional neural networks. *Language Resources and Evaluation* (Sep 2019). 72, 73

[146] Wu, F., Fan, A., Baevski, A., Dauphin, Y. N., and Auli, M. Pay less attention with lightweight and dynamic convolutions. *CoRR abs/1901.10430* (2019). 1

[147] Wu, L., Petroni, F., Josifoski, M., Riedel, S., and Zettlemoyer, L. Zero-shot Entity Linking with Dense Entity Retrieval. *arXiv e-prints* (Nov. 2019), arXiv:1911.03814. 114, 119, 121, 122

[148] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR abs/1609.08144* (2016). 25

[149] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 5753–5763. 1

[150] Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Cambridge, MA, USA, 2015), NIPS, MIT Press, pp. 649–657. 43, 44

Appendix A

Proposal for low-resource entity linking

We originally planned to explore entity linking in low-resource scenarios as a contribution to the KnEDLe project. But due to computational power and time constraints, we opted instead to contribute to extending the VICTOR dataset, which resulted in the work we present in §4.3. In this chapter we describe our research plan for low-resource entity linking and leave its execution for future work.

A.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 [71]. 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 [82].

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 ($\approx 80\%$ [112]) can be solved by a baseline that, given a mention m, chooses the entity e that maximises $p(e|m)$. 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 presents a research proposal with aims to develop 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 A.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;

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

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 (§A.2), we examine recent research on EL. Then (§A.3) we detail what we want to achieve, how we intend to do it, and when we expect to conclude each step.

A.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 A.1 summarises our analysis.

Table A.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. [138]	2016		✓	✓		
Ganea et al. [38]	2017		✓	✓		✓
Pappu et al. [103]	2017	✓	✓	✓		✓
Upadhyay et al. [140]	2018		✓	✓	✓	
Kolitsas et al. [71]	2018	✓	✓	✓		✓*
Gillick et al. [41]	2019				✓	✓
Le et al. [78]	2019				✓	
Logeswaran et al. [82]	2019					✓
Le et al. [77]	2019		✓	✓	✓	✓*
Broscheit [16]	2019	✓				
Wu et al. [147]	2019					✓
Onoe et al. [101]	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 [82]. **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.

A.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 [129, 84], 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 [71].

Pappu et al. [103] 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. [71] 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 [16] 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 A.2 illustrates the approach. Broscheit attached an output classification layer on top of BERT [31] and trained the architecture on Wikipedia text data. Though the method did not outperform the one proposed by Kolitsas et al. [71], it is free from the entity dictionary and frequency statistics assumptions the latter relies on.

A.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 [38]. Local context is

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

Table A.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 [16].

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

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 [138] engineered two features that capture global context: `other-mentions`(m), a set of vectors that represent the other mentions in the document; 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 [38] 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 [77] 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 [53] 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. [140] 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. [103] 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. [71] 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.

A.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 [138] 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 [38, 140, 71, 77] employed $p(e|m)$ both for CG and as a feature for ED.

Pappu et al. [103] 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 [101] 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.

A.2.4 Structured data

Relationship tuples and entity type annotation can be used to improve ED [82]. 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. [140] 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. [41] used Wikipedia categories as one of the sources of information for entity encoding. When T-SNE [141] 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 [78] 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

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

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 [77] 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 [101] 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%).

A.2.5 Entity dictionary

Most works we studied assumed the existence of an entity dictionary $\mathcal{E} = \{(e_i, d_i)\}_{i=1,\dots,k}$ 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 [38] 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. [103] 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. [41] 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. [82] and Wu et al. [147] both employed BERT [31] 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

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

where [CLS], [M_s], [M_e] and [SEP] are special tokens: the context-candidate embedding is given by last layer of the output of [CLS]; [M_s] and [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. [71] and Le et al. [77] indirectly assumed an entity dictionary since they borrowed the entity embeddings trained by Ganea and Hofmann [38]. 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.

A.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.

A.3.1 Modelling

Broscheit's work [16] 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.

[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.

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 [82, 147], where the scientists used a model pre-trained on large corpora [31] and then fine-tuned it on the zero-shot dataset introduced by Logeswaran et al. [82]. 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 [38] or feeding an entity and the most common words found near its mentions to a transformer [82, 147].

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 [126]. 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:

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

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 [97] 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 [96] and fasttext [10] 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.

A.3.2 Datasets

In this subsection we introduce some corpora with EL annotation.

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

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. [147] contains 9 million mentions and 5.9 million entities.

Wikia zero-shot corpus The zero-shot EL dataset proposed by Logeswaran et al. [82] 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 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 [62] 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 [53], the original WikilinksNED dataset [35], and the Unseen-Mentions version created by Onoe and Durret [101].

A.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

$$\text{Recall@k} = \frac{n}{m}\,. \tag{A.3}$$

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

$$\text{Unnormalised accuracy} = \frac{c}{m}\,. \tag{A.4}$$

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

$$\text{Normalised accuracy} = \frac{d}{n} \tag{A.5}$$

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