UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica e de Computação

Guilherme Moraes Rosa

Exploring Zero-shot Models for Cross-lingual and Cross-domain Transfer Learning

Explorando Modelos Zero-shot para Transferência de Conhecimento Multilíngue e entre Domínios

Campinas

2022

Guilherme Moraes Rosa

Exploring Zero-shot Models for Cross-lingual and Cross-domain Transfer Learning

Explorando Modelos Zero-shot para Transferência de Conhecimento Multilíngue e entre Domínios

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Computer Engineering.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Engenharia de Computação.

Supervisor: Prof. Dr. Roberto de Alencar Lotufo
Co-supervisor Dr. Rodrigo Frassetto Nogueira

Este trabalho corresponde à versão final da dissertação defendida pelo aluno Guilherme Moraes Rosa, e orientada pelo Prof. Dr. Roberto de Alencar Lotufo.

Campinas
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Elizangela Aparecida dos Santos Souza - CRB 8/8098

R71e

Rosa, Guilherme Moraes, 1989-
Exploring zero-shot models for cross-lingual and cross-domain transfer learning / Guilherme Moraes Rosa. – Campinas, SP : [s.n.], 2022.

Orientador: Roberto de Alencar Lotufo.
Coorientador: Rodrigo Frassetto Nogueira.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Inteligência artificial. 2. Aprendizado profundo. 3. Processamento de linguagem natural (Computação). 4. Redes neurais (Computação). 5. Transferência de aprendizagem. I. Lotufo, Roberto de Alencar, 1955-. II. Nogueira, Rodrigo Frassetto, 1986-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Explorando modelos zero-shot para transferência de conhecimento multilíngue e entre domínios
Palavras-chave em inglês:
Artificial intelligence
Deep learning
Natural language processing
Neural networks
Transfer learning
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Roberto de Alencar Lotufo [Orientador]
Viviane Pereira Moreira
Hélio Pedrini
Data de defesa: 09-03-2022
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-7510-2621
- Currículo Lattes do autor: http://lattes.cnpq.br/3377082115019979

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Guilherme Moraes Rosa **RA:** 264437
Data da defesa: 09/03/2022

Dissertation Title: "Exploring Zero-shot Models for Cross-lingual and Cross-domain Transfer Learning".

Titulo da Dissertação: "Explorando Modelos Zero-shot para Transferência de Conhecimento Multilíngue e entre Domínios".

Prof. Dr. Roberto de Alencar Lotufo (Presidente, FEEC/UNICAMP)
Dra. Viviane Pereira Moreira (INF/UFRGS)
Prof. Dr. Hélio Pedrini (IC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. Lotufo and my co-supervisor Dr. Nogueira for providing me with such an incredible opportunity. This period working with you has been one of great learning and self-improvement, you two are role models that inspire me to always keep pushing my limits. In particular, I would like to highlight my appreciation for Dr. Nogueira, who guided me in such a positive way and who always made me feel confident in my abilities. Also, I would like to thank NeuralMind for funding my research and for letting me be a part of this amazing team.

I especially thank my mother Marilene, without your support I would not be able to pursue my goals and without you I would not have been able to get here. I would like to thank my girlfriend Michele for being so patient all these years and for helping me for hours on end whenever I needed it. You are just perfect and your encouragement is essential. I also thank my aunt Maria José for the support and unconditional trust since my childhood, I will make you proud.

To conclude, I cannot fail to thank the special people I met and who somehow were important at some point in my life, inspiring and helping me along this journey.

Science is the great antidote to the poison of enthusiasm and superstition."
(Adam Smith)

Abstract

Deep learning algorithms have been adopted in many important applications in natural language processing. These algorithms stand out for their ability to learn large amounts of information and perform well on tasks that were previously considered too difficult for machines to perform. Therefore, its application has been increasingly widespread for different tasks, domains and languages. Yet, it is well known that deep learning models typically do not generalize much beyond the data distribution seen during fine-tuning and have difficulty adapting to new scenarios. A solution to this problem is to retrain the model on a new large and diverse labeled dataset. However, we often do not have readily available datasets for every new scenario that may arise, and in addition, real-world data is constantly changing. Thus, an effective method to address this problem and improve the generalization capacity of transformer models is to use zero-shot transfer learning approaches.

To further study the transfer learning ability of transformer models, we separate zero-shot learning into two different categories, depending on how the test examples differ from the data used for fine-tuning. In our work, training and test examples may differ because they belong to different languages (cross-lingual) or to different domains (cross-domain). We explore both categories by designing two studies that cover each separately. In our first study, we analyze three cross-lingual methods in terms of their effectiveness (e.g., accuracy), development and deployment costs, as well as their latencies at inference time. Furthermore, by combining cross-lingual methods, we achieve the state of the art in two datasets used in the first study. In our cross-domain study, we investigate the transfer learning ability from general domain to the legal domain. For that, we participated in COLIEE 2021, a competition involving automated tasks in the legal domain, in which we experimented with transformer models with no adaptations to the target domain. Our submissions to the task of legal case entailment achieved the highest scores, surpassing the second-best team by more than six points and our zero-shot model outperformed all fine-tuned models on this task. In addition, our experiments confirm a counter-intuitive result in the new paradigm of pretrained language models: given limited labeled data, models with little or no adaptation to the target task can be more robust to changes in the data distribution than models fine-tuned on it.

Keywords: Natural language processing; Zero-shot; Cross-lingual; Transfer learning; Machine learning; Legal NLP; Legal case entailment.

Resumo

Os algoritmos de aprendizado profundo têm sido adotados em diversas aplicações importantes no processamento de linguagem natural. Esses algoritmos se destacam por sua capacidade de aprender grandes quantidades de informações e atingir ótimos desempenhos em tarefas antes consideradas muito difíceis de serem realizadas por máquinas. Portanto, sua aplicação tem sido cada vez mais difundida para diferentes tarefas, domínios e idiomas. Ainda assim, sabe-se que modelos de aprendizado profundo normalmente não generalizam muito além da distribuição de dados vista durante o treinamento e têm dificuldade em se adaptar a novos cenários. Uma solução para este problema é treinar novamente o modelo em um novo conjunto de dados rotulado grande e diverso. No entanto, muitas vezes não temos conjuntos de dados prontamente disponíveis para cada novo cenário que possa surgir e, além disso, dados do mundo real estão em constante mudança. Assim, um método eficaz para resolver este problema e melhorar a capacidade de generalização de modelos transformer é usar abordagens de transferência de conhecimento *zero-shot*.

Para estudar com maior profundidade a capacidade de transferência de conhecimento de modelos *transformer*, separamos o aprendizado *zero-shot* em duas categorias diferentes dependendo de como os exemplos de teste diferem dos dados usados para treinamento. Em nosso trabalho, os exemplos de treinamento e teste podem ser diferentes por pertencerem a idiomas diferentes (*cross-lingual*) ou a domínios diferentes (*cross-domain*). Exploramos ambas as categorias projetando dois estudos que cobrem cada uma separadamente. Em nosso primeiro estudo, analisamos três métodos de transferência de conhecimento entre diferentes idiomas em termos de eficácia (por exemplo, acurácia), custos de desenvolvimento e implantação, bem como suas latências em momento de inferência. Além disso, ao combinar métodos de transferência multilíngue, alcançamos o estado da arte em dois conjuntos de dados utilizados neste primeiro estudo. Em nosso estudo *cross-domain*, investigamos a capacidade de transferência de conhecimento do domínio geral para o domínio jurídico. Para isso, participamos do COLIEE 2021, competição que envolve a execução de tarefas automatizadas aplicadas ao domínio jurídico, no qual experimentamos modelos *transformer* sem adaptações ao domínio alvo. Nossas submissões para a tarefa de vinculação de processos judiciais obtiveram as pontuações mais altas, ultrapassando a segunda melhor equipe em mais de seis pontos e nosso modelo *zero-shot* superou todos os modelos treinados para esta tarefa. Além disso, nossos experimentos confirmam um resultado bastante contra-intuitivo no novo paradigma de modelos de linguagem pré-treinados: dada uma limitação na quantidade de exemplos rotulados, modelos com pouca ou nenhuma adaptação à tarefa alvo podem ser mais robustos a mudanças na distribuição de dados do que modelos diretamente treinados no conjunto de dados alvo.

Palavras-chaves: Processamento de linguagem natural; Multilíngue; Transferência de

conhecimento; Aprendizado de máquina; PLN no domínio jurídico; Vinculação de processos legais.

List of Figures

List of Tables

List of Acronyms

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
API	Application Programming Interface
ASSIN2	Segunda Avaliação de Similaridade Semântica e Inferência Textual
BERT	Bidirectional Encoder Representation from Transformers
BM25	Best Match 25
C4	Colossal Clean Crawled Corpus
CBOW	Continuous Bag of Words
COLIEE	Competition on Legal Information Extraction/Entailment
DeBERTa	Decoding-enhanced BERT with Disentangled Attention
e.g.	exemplum gratia (en: for example)
et al.	et alia (en: and others)
GermanQuAD	German Question Answering Dataset
GPT-3	Generative Pre-Training Transformer 3
GPU	Graphic Processing Unit
IBM	International Business Machines
ICAIL	International Conference on Artificial Intelligence and Law
i.e.	id est (en: that is)
LM	Language Model
LSTM	Long Short-Term Memory
mBERT	Multilingual Bidirectional Encoder Representation from Transformers
mC4	Multilingual Colossal Clean Crawled Corpus
MLM	Masked Language Modeling
mMARCO	Multilingual Machine Reading Comprehension
MNLI	Multi-Genre Natural Language Inference
MRR	Mean Reciprocal Rank
MS MARCO	Microsoft Machine Reading Comprehension
mT5	Multilingual Text-to-Text Transfer Transformer
NAACL	North American Chapter of the Association for Computational Linguistics
nDCG	Normalized Discounted Cumulative Gain
NER	Named Entity Recognition
NewsQA	News Question Answering
NLI	Natural Language Inference

NLP	Natural Language Processing
NMT	Neural Machine Translation
NSP	Next Sentence Prediction
POS	Part of speech
QA	Question Answering
SQuAD	Stanford Question Answering Dataset
T5	Text-to-Text Transfer Transformer
TPU	Tensor Processing Unit
TREC	Text Retrieval Conference
USD	United States Dollar
ViQuAD	Vietnamese Question Answering Dataset
XLM	Cross-Lingual Language Model
XLM-R	Cross-Lingual Language Model Roberta
XNLI	Cross-lingual Natural Language Inference
XQuAD	Cross-lingual Question Answering Dataset

Contents

1 Introduction

The development of machines capable of simulating human intelligence and performing tasks historically done by humans, such as speech and facial recognition or decision making, is a goal that researchers have pursued for decades. Although we are not close to reaching this goal yet, it has driven research and, in recent years, deep learning models have stood out in solving complex problems once considered too difficult to be solved by machines. Especially for tasks where the inputs are unstructured data, i.e., the data is not in tabular format, but images, text or audio data, these models excel at learning from huge amounts of data and can achieve great performance in complex applications such as personal assistants, self-driving cars and medical diagnostics. Because of this capability, deep learning has proven to be useful in many industry sectors and companies are increasingly relying on learning algorithms. This technology is applied to manufacturing, finance, law, healthcare and many other sectors of the economy and is behind visual recognition and recommendation systems, chatbots, language translation apps, fraud detection, social media feeds, etc.

Despite recent successes, it has been observed that deep learning models typically do not generalize much beyond the data seen during fine-tuning and have difficulty adapting to new scenarios (RAMPONI; PLANK, 2020; NOZZA *et al.*, 2016; WU *et al.*, 2020). To overcome this, a large enough labeled dataset containing relevant examples of the new scenario is often needed. However, we do not have readily available data for every new scenario that may arise, and real-world data is constantly changing, making it impractical to frequently retrain a model for each new scenario and increasing the need for models with strong generalization capabilities. In the field of natural language processing (NLP), this problem arises from the low availability of high-quality datasets for fine-tuning (HUANG *et al.*, 2019) in low-resource languages and technical text domains, as data annotation is costly both in terms of money and time spent. An effective alternative to address this problem and improve the generalization ability of deep learning models, whether to a new task, domain or language, is to use a zero-shot transfer learning approach. Its goal is to transfer knowledge of related languages, tasks or domains to a target scenario whose specifications were unknown during the development of the model. This type of approach is important because it allows us to develop models with good performance for situations in which labeled data is scarce.

In this dissertation, we separate zero-shot learning into two different categories depending on the test data provided and how it distinguishes from the training data. Training and test examples may differ in belonging to different languages (cross-lingual) or different domains (cross-domain). We explore these categories by designing separate

studies that cover cross-lingual and cross-domain transfer learning to answer the following question: given the availability of large supervised datasets in English and high performance transformer models pretrained on general domain text, what is the most effective way to use these resources in limited labeled data scenarios, especially for low-resource languages and legal texts? The answer to this question allows us to effectively develop and deploy natural language processing systems for tasks where there is not sufficient labeled data to fine-tune the models.

The first study falls in the cross-lingual category and addresses an important problem in NLP, which is the low availability of methods, datasets and models in the vast majority of languages spoken today. Most research has focused on advancing methods that work well only for a small number of high-resource languages, for example English. This discrepancy between the number of languages spoken and the reality of research has major practical implications for the development of high-performance models in low-resource languages, in addition to systematically excluding speakers of these languages from access to NLP-based technologies.

In contrast to this problem, recent work demonstrates that multilingual pretrained transformer models show great cross-lingual zero-shot performance, i.e., models fine-tuned on a dataset of a high-resource language perform well on the same task in another language (WU; DREDZE, 2019; CONNEAU *et al.*, 2020; XUE *et al.*, 2021). The current literature, however, focuses mainly on developing transfer learning methods that lead to a better model with respect to some performance metrics, ignoring important decision-making parameters, such as development and deployment costs. Considering the relevant associated costs, the cross-lingual study analyzes the feasibility and cost-effectiveness of three cross-lingual transfer learning methods: 1) fine-tuning a model on a source language and evaluating it on the target language without translation, i.e., in a zero-shot manner; 2) automatic translation of the training dataset to the target language; 3) automatic translation of the test set to the source language at inference time and evaluation of a model fine-tuned on the source language.

Although there has been extensive research about the performance of pretrained language models in general domain texts (e.g., from the web), the application of these models to technical or specialized documents, with some exceptions, has received less attention. Therefore, in the cross-domain study, we explore the ability to transfer knowledge in a zero-shot manner from a general domain to a specific domain.

It is well known that domain-specific models can achieve great performance given enough in-domain data. However, in many specific text domains, a sufficiently large annotated dataset is not available, such as technical manuals and medical or legal documents, as collecting and labeling such data is expensive, time-consuming, and requires specialized knowledge. Nevertheless, it has been observed that pretrained language models fine-tuned

only on a large and diverse supervised dataset have shown strong zero-shot capabilities (THAKUR *et al.*, 2021), i.e., they can transfer well to a variety of out-of-domain tasks. A zero-shot multi-stage retrieval pipeline based on the T5 model (RAFFEL *et al.*, 2020) was the first or second best performing system in 4 tracks of the TREC 2020 competition (PRADEEP *et al.*, 2020), including domain-specific tasks (ROBERTS *et al.*, 2019; ZHANG *et al.*, 2020). Despite previous studies in other domains, as far as we know, to date, there is no strong evidence that zero-shot models transfer well to the legal domain, as most of state-of-the-art models need adaptations to the target task. In this study, we show that for the legal case entailment task, our zero-shot model (fine-tuned only on general domain texts) performs better than the models fine-tuned on the task itself.

1.1 Objectives

This dissertation studies the problem of learning representations that transfer well across languages and text domains using transformer-based models for natural language processing. The main hypothesis is the following:

In a limited labeled data scenario, a zero-shot model can outperform models directly fine-tuned on the target task.

We mainly address the problem of limited labeled data in key applications, emphasizing the low availability of large supervised datasets and high-performance transformer models, both for most languages and for technical domain texts. Our goal is to investigate the transfer learning capabilities of transformer models, especially in zero-shot scenarios. First, we explore the cross-lingual capability of transformer models to perform well in three different languages, including low-resource languages such as Portuguese and Vietnamese. Then, in the cross-domain study, we explore the model's ability to transfer knowledge from general domain texts to legal domain texts, which is a very technical textual genre, that has its own particularities and complexities, in addition to being quite different from a general domain text.

1.2 Contributions

The key contributions of this work address a similar issue in two different categories of zero-shot learning, which is the small amount of data available for certain scenarios. Our main contribution is to develop zero-shot models that are competitive or even outperform existing fine-tuned models in the target language and target domain, reducing the need for annotated data.

Our contribution in the cross-lingual study is to evaluate knowledge transfer methods while also considering their financial and computational costs. In addition, we compare two translation approaches in three different tasks. We also show that automatically translating question answering datasets is not trivial and propose a new method for translating them. Finally, while exploring the best cross-lingual methods, we achieve the state of the art on two datasets in a low-resource language, thus showing that our cross-lingual methodology is sound.

Our contribution in the cross-domain study is to investigate the zero-shot transfer ability to the legal domain. For that, we participated in the legal case entailment task of COLIEE 2021 (RABELO *et al.*, 2021), in which our transformer model with no adaptation to the target domain outperformed all fine-tuned models. Our result confirms, in the legal domain, a counter-intuitive recent finding in other domains: that given limited labeled data, zero-shot transformer models tend to perform better on held-out datasets than models fine-tuned directly on the target task. In addition, we also show that the performance of our zero-shot approach can be further improved by scaling to a 3 billion parameter model.

1.3 Dissertation structure

This dissertation is organized as follows: in Chapter 2, we review relevant concepts used throughout this work, such as the transformer architecture, transfer learning and related topics. In Chapter 3, we describe our cross-lingual study, its analysis and results. Then, in Chapter 4, we describe our cross-domain experiments and present its results. Lastly, we make our conclusions in Chapter 5.

1.4 Publications and awards

In this section, we present the publications and awards related to this dissertation, as well as the articles published during the Master's.

Our cross-lingual study (ROSA *et al.*, 2021a) was published as a preprint paper and will soon be submitted to a future conference.

- Rosa, G., Bonifácio, L.H., Souza, L., Nogueira, R., Lotufo, R. (2021). *A cost-benefit analysis of cross-lingual transfer methods*. arXiv preprint arXiv:2105.06813.

Our cross-domain study (ROSA *et al.*, 2021c) was presented at the 18th International Conference on Artificial Intelligence and Law (ICAIL) and published in the respective

proceedings. The models resulting from this work reached the top three places in the task of legal case entailment in the COLIEE 2021 competition.

- Rosa, G., Rodrigues, R., Nogueira, R., Lotufo, R. (2021). *To Tune or Not To Tune? Zero-shot Models for Legal Case Entailment*. ICAIL'21, Eighteenth International Conference on Artificial Intelligence and Law, June 21–25, 2021, São Paulo, Brazil.

The article about our submission to the task of legal case retrieval (ROSA *et al.*, 2021b) in the COLIEE 2021 competition was presented at the COLIEE workshop held at ICAIL 2021 and published in the Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, in which we won second place.

- Rosa, G., Rodrigues, R., Nogueira, R., Lotufo, R. (2021). *Yes, BM25 is a Strong Baseline for Legal Case Retrieval*. Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment.

Finally, although not directly related to this dissertation, the following article (PIRES *et al.*, 2022) was also published over the course of the Master's.

- Pires, R., Souza, F., Rosa, G., Nogueira, R., Lotufo, R. (2022). *Sequence-to-Sequence Models for Extracting Information from Registration and Legal Documents*. arXiv preprint arXiv:2201.05658.

2 Background

This chapter introduces background knowledge to prepare the reader for subsequent chapters. It starts by providing an overview of the fundamentals of artificial intelligence and its respective subfields related to this work, namely machine learning, deep learning and natural language processing. Next, we delve into the transformer architecture, as it is the cornerstone of all models used throughout this work. Finally, in the sections on transfer learning, zero-shot learning, cross-lingual learning and cross-domain learning, we set the stage for the next chapters, where we study and apply transfer learning methods to leverage the knowledge of our models across different tasks, languages and domains.

2.1 Artificial intelligence

Artificial intelligence (AI) is the field that attempts to develop computer systems capable of simulating human intelligence (RUSSELL; NORVIG, 2020). Some of the main challenges faced by the area is to provide machines with some human characteristics such as knowledge and reasoning, problem solving, perception, learning, planning and the ability to manipulate objects. The goal of AI is to create machines capable of learning from experience and exhibiting intelligent human-like behavior. This means machines that can understand text written in natural language, use vision to recognize people, scenes or objects, make intelligent decisions and interact with the physical world (PIETIKäINEN; SILVEN, 2022).

Artificial intelligence was founded as a research discipline in the 1950's, based on the assumption that human intelligence can be accurately described and therefore simulated by machines. During all that time, the field has experienced several waves of optimism and disappointment, followed by increased interest in recent decades as mathematical statistical machine learning has dominated the field, proving to be very successful in solving a wide range of challenging problems (TOOSI *et al.*, 2021). Artificial Intelligence and its respective subfields have been advancing rapidly and being increasingly developed in recent years, bringing intricate and complex concepts such as machine learning, deep learning and natural language processing. Figure 1 shows how Artificial intelligence and its subfields are related.

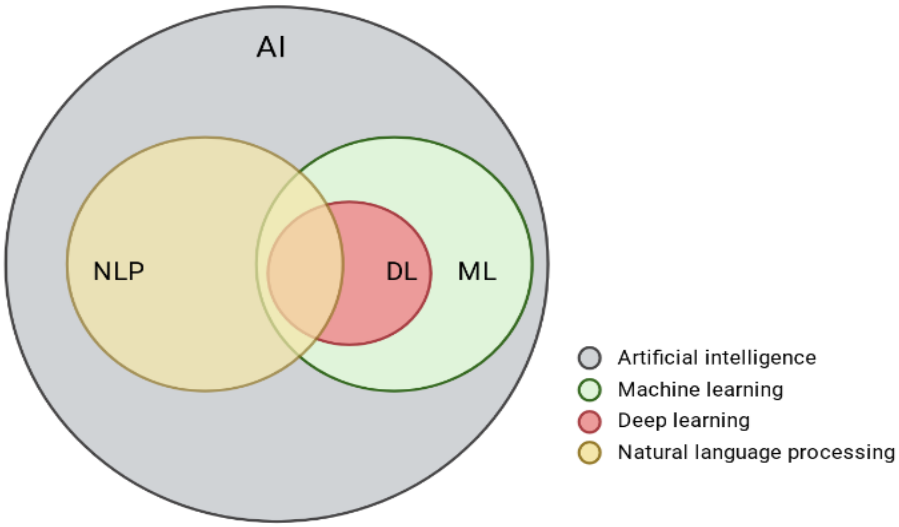

Figure 1 – Artificial intelligence and its subfields.

2.2 Machine learning

Machine learning is a subfield of artificial intelligence whose systems are capable of automatically learning from experience, without the need for explicit programming (WANG; RAJ, 2017). Compared to modern deep learning approaches, classical machine learning algorithms rely on more human intervention to learn, as experts need to determine the features that best describe the inputs, usually structured data in a tabular format, to generate good predictions.

We can divide machine learning into two broad categories based on how feedback is given to the model during the learning process. In supervised learning, an algorithm learns from datasets in which a human has carefully selected desired inputs and outputs. During training, a machine learning model compares its predictions to the actual outputs so it can understand the underlying data structure and learn a function that fits the data. In unsupervised learning, an algorithm analyzes seemingly unrelated data in an unlabeled dataset and must find hidden patterns in the underlying data structure to meaningfully organize it. This category of algorithms is valuable because unlabeled data is abundant and no human supervision is required.

2.3 Deep learning

Deep learning is a branch of machine learning that focuses on training deep artificial neural networks on huge datasets to solve a large number of tasks (SCHMIDHUBER, 2014). A deep neural network is an interconnected web of millions of neurons in multiple layers designed to mimic information processing in the human brain. These deeper neural networks have the ability to learn from large amounts of data, improving their performance as more data is provided, which is profoundly different from classic machine

learning techniques that stop improving performance by hitting a plateau.

In addition to scalability, another important improvement of deep learning models is the ability to perform feature learning, i.e., automatically extract relevant features from raw data to learn good representations. This capability allows a model to learn complex functions that map input to output without the need for human-crafted features.

Deep Learning is widely applied to predictive analytics, computer vision and natural language processing tasks (HEATON, 2020).

2.4 Natural language processing

Natural language processing is the area of research that studies and develops intelligent machines capable of simulating the human ability to use natural language through text or speech, that is, to communicate in human languages such as English, Portuguese or Vietnamese (TORFI *et al.*, 2020; OTTER *et al.*, 2018). The field derives from many disciplines, for example computer science and computational linguistics, in an attempt to bridge the communication gap between humans and computers. NLP includes the subareas of natural language understanding and natural language generation, which attempts to mimic the human ability to read, understand, and generate natural language for a wide range of tasks and applications, such as answering questions, summarizing texts, chat bots and virtual assistants. Current NLP systems are capable of analyzing large amounts of text, understanding complex textual concepts, and deciphering language ambiguities to extract useful information, relationships, or even provide summaries. Given the huge amount of unstructured data that we have available today, mainly on the internet and social media, NLP systems have become essential for efficiently analyzing data.

Until the 1990s, most natural language processing systems used complex hand-written rules. But since then, with a novel approach based on machine learning combined with greater computing power, a new generation of algorithms has emerged. These statistical machine learning models were able to make relatively good predictions based on probabilistic decisions.

The first neural language model was proposed in 2001 (BENGIO *et al.*, 2001) and used a feed-forward neural network. In this type of network, data only moves in one direction, from input neurons, through any hidden neurons, and then to the output. The model is trained to predict the next word in a sentence based on previous words, a task we call language modeling. Despite its simplicity, language modeling is essential for developing today's state-of-the-art models, as they all rely on a form of language modeling in their pretraining objective.

Another important milestone in NLP is the development of dense vector representations, i.e., word embeddings. Unlike sparse vector representations or, as it is better

known, bag-of-words model, this new technique represents words as real-valued vectors in a predefined vector space. A word is mapped to a vector and the vector values are learned using an optimization strategy. Training word embeddings in large corpora allows vectors to capture the semantic and syntactic characteristics of words and can be done through two training objectives. One is to predict the surrounding words given the word in the center (Skip-gram) and the other is to predict the word in the center given the surrounding words (CBOW). Thus, these learned representations can approximate certain relationships between words such as gender, country-capital and words that have the same meaning often have a similar representation. Two successful examples of word embedding are word2vec (MIKOLOV *et al.*, 2013) and GloVe (PENNINGTON *et al.*, 2014). A promising research direction is to develop word embeddings capable of placing similar words, but in different languages, into similar representations located close together in the same vector space. The aim is to improve cross-lingual transfer, including zero-shot, especially for low-resource languages.

Although recurrent neural networks started to be adopted in natural language processing around 2013, the concept of RNN dates back to 1986 (JORDAN, 1986) and the long short-term memory (LSTM) architecture was developed in 1997 (HOCHREITER; SCHMIDHUBER, 1997). RNNs are a class of neural networks that can be seen as an extension of feed-forward neural networks, since RNNs have an additional presence of feedback loops in the hidden layers, which allows the network to identify the spatial position of words in a sentence and makes this architecture a natural choice for processing sequential data such as text. Long short-term memory networks are a variation of RNNs designed to deal with the vanishing gradient problem that occurs in the process of fine-tuning recurrent neural networks using backpropagation. During fine-tuning, each of the neural network weights receives an update proportional to the error between the predictions and the ground-truth values, which we call a gradient as it is calculated using partial derivatives. The problem is that in some situations, due to the calculations involved, the gradients that are being backpropagated can tend to zero (vanish), preventing the weights from changing their values and completely interrupting the learning process of the neural network. LSTM solves this problem by using special memory cells capable of selecting which information is important to remember or which can be forgotten.

After a successful application of LSTMs in a wide range of NLP tasks, an end-to-end approach to sequence learning was proposed (SUTSKEVER *et al.*, 2014), a new architecture composed of two LSTM networks combined. In the proposed approach, the first LSTM (encoder) processes an input sentence sequentially and compresses it into a vector representation, then a second LSTM (decoder) predicts the output sentence word by word based on the encoder representation. The sequence-to-sequence architecture ended up being applied mainly to the task of machine translation achieving good results. Following these results, a fundamental improvement was proposed that allowed

neural machine translation models to outperform classical systems, the attention mechanism (BAHDANAU *et al.*, 2016). Attention enables a model to automatically give more importance to the parts of an input sentence that are most relevant to help predict the correct target sentence.

The transformer architecture builds on many of these earlier contributions, such as word embedding, sequence-to-sequence and attention mechanism. Especially the idea of attention is a central element in modern NLP, where models use multiple layers of attention to look at the surrounding words in a sentence to get more contextually sensitive word representations to achieve the state of the art in several NLP tasks.

2.5 Transformers

This section provides an overview of the transformer architecture proposed in 2017 (VASWANI *et al.*, 2017). All models used in this work are based on this architecture shown in Figure 2 with minor variations.

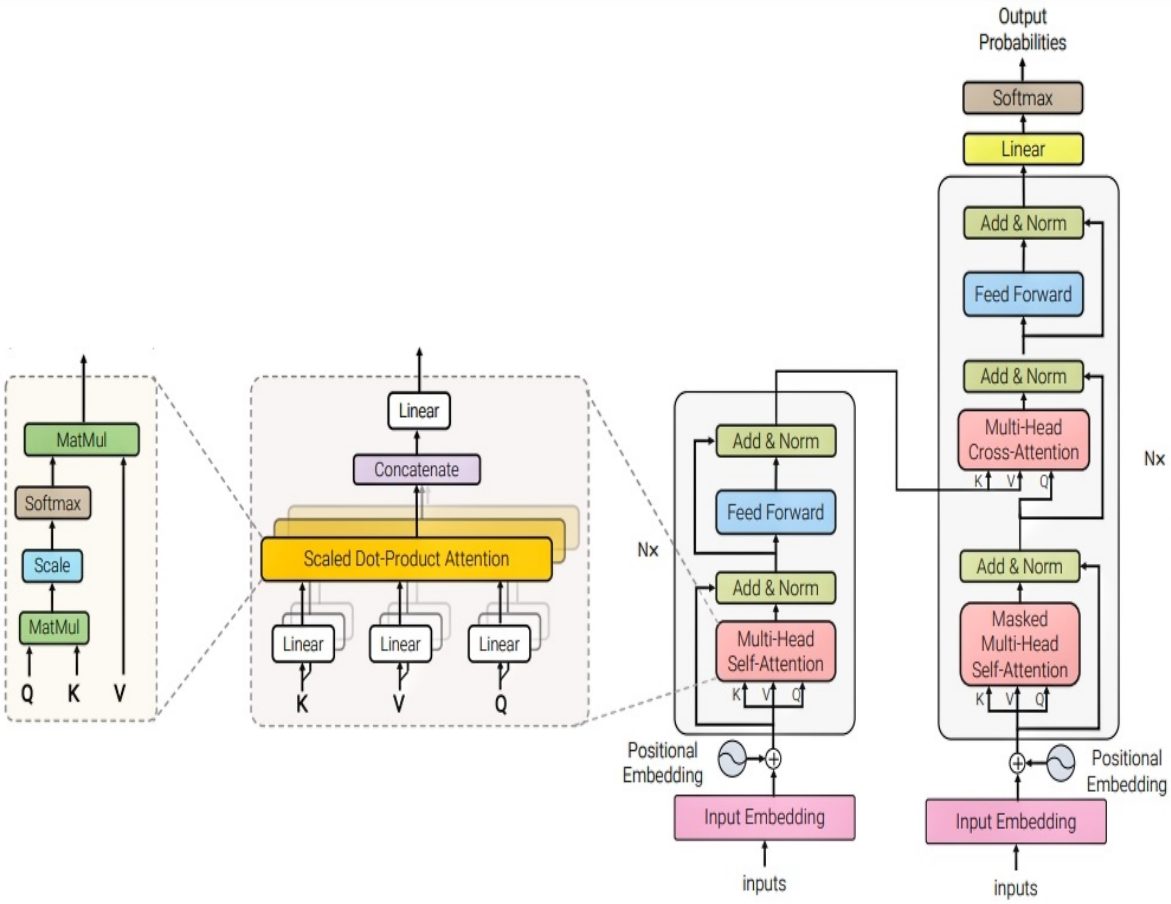

Figure 2 – Transformer architecture. Figure adapted from "Efficient Transformers: A Survey" by Tay et al. (2020).

Transformer is a deep learning model originally designed to handle sequential data, mainly used in the area of natural language processing, capable of weighing the in-

fluence of different parts of the input data by applying a mechanism called self-attention. Transformers-based models quickly became ubiquitous for NLP problems, replacing models such as LSTMs.

2.5.1 Tokenization

The tokenization process consists of breaking a text into smaller units and converting it into a sequence of known tokens that belong to a predefined vocabulary. Tokenization is mainly performed at the word or subword level. Some examples include the WordPiece (SCHUSTER; NAKAJIMA, 2012) and SentencePiece (KUDO; RICHARDSON, 2018) algorithms.

2.5.2 Embeddings

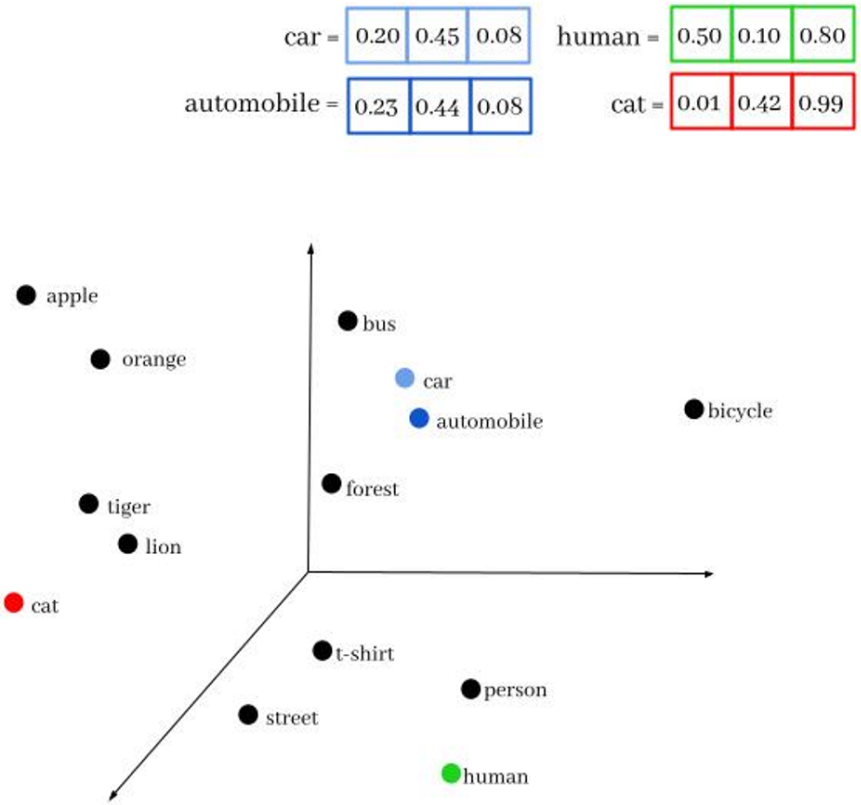

Figure 3 – Word embeddings in a vector space of three dimensions. Words with similar meaning should be located closer.

In natural language processing, word embedding (MIKOLOV *et al.*, 2013; MIKOLOV, 2013) is a learned representation from text that encodes the meaning of a word into an n-dimensional real-valued vector. As shown in Figure 3, words with close meaning are expected to have a similar representation and be located closer in vector space. Transformer

learns its own word embeddings from scratch, starting from a random initialization and refining them during pretraining.

2.5.3 Attention

One of the main innovations in the transformer architecture is the extensive use of attention mechanisms (BAHDANAU *et al.*, 2016), as it allows the model to focus on closely related words in the input sentence. Transformer improves the implementation of attention mechanisms, previously used for machine translation in architectures based on RNNs (LUONG *et al.*, 2015), by removing recurrence and relying primarily on self-attention, where the representation of each word in a sentence is calculated by relating all the words in the same sentence. Basically, attention is used in three places in the transformer architecture: self-attention at the encoder, self-attention at the decoder and the encoder-decoder-attention at the decoder.

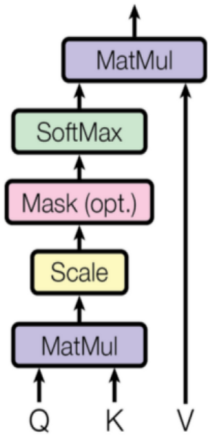

Figure 4 – Scaled dot-product attention. Figure from "Attention Is All You Need" by Vaswani et al. (2017).

Figure 4 shows the basic building blocks of the transformer attention mechanism, the scaled dot-product attention units or, as it is better known, the self-attention mechanism. These units produce weighted embeddings for each token in a sentence containing combined information from the token itself and other tokens relevant to the context. First, the attention layer takes three parameters as input, known as query, key, and value. In an efficient implementation, using batches, these parameters are matrices created by multiplying the word embeddings by three matrices fine-tune throughout the training process. At the end, we have query, key and value projections for each word in the input sentence. The second step is to calculate a score for each word in the input sentence against all other words in the sentence. The score is calculated by taking the dot product of the query matrix with the key matrix and determines how much focus to put on other parts of the input sentence as we encode a word at a given position. In the third step the scores are divided by the square root of the dimension of a key vector, in the fourth step the

previous result is passed through a softmax function to obtain positive scores that add up to 1. The softmax result determines how much each word in a sentence will be expressed in a particular word embedding. The fifth step consists of multiplying the value matrix by the softmax scores to keep the values of the words we want to focus on and decrease the values of the irrelevant words. The attention unit output is the weighted sum of the value matrices of all tokens.

Formally, let $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ be the query, key and value matrices. The matrix of outputs is computed by:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.1}$$

2.5.3.1 Self-attention in the encoder

In the encoder self-attention, the input sequence pays attention to itself and is passed to all three parameters, query, key, and value.

2.5.3.2 Self-attention in the decoder

In the self-attention mechanism present in the decoder, the target sequence pays attention to itself and is also passed to all three parameters, query, key, and value.

2.5.3.3 Encoder-decoder-attention

In the encoder-decoder attention mechanism, the target sequence pays attention to the input sequence. The output of the decoder is passed to the query parameter, while the output of the last encoder is passed to the value and key parameters.

2.5.4 Multi-head attention

In transformer architecture, one set of matrices (query, key, value) is called an attention head and each layer has multiple attention heads. As shown in Figure 5, the attention module splits its query, key, and value parameters multiple times and passes each split independently through a separate head. The multiple outputs of the multi-head attention are combined to produce a final attention score and then move on to the next feed-forward layer.

2.5.5 Positional encoding

As the attention mechanism does not take into account the position of words, some information about the relative or absolute position of the tokens must be included in the input. Positional encoding is the approach to considering the order of words in the

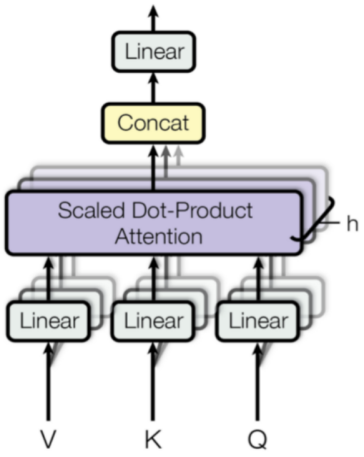

Figure 5 – Multi-head attention. Figure from "Attention Is All You Need" by Vaswani et al. (2017).

input sequence. It simply consists of adding a vector to each input embedding to help the model determine the position of each word or the distance between different words in the sequence. The original transformer uses fixed absolute encodings given by

$$PE(\text{pos}, 2i) = sin(\text{pos}/10000^{2i/H})$$
$$PE(\text{pos}, 2i+1) = cos(\text{pos}/10000^{2i/H})$$

where $pos \in \{1, \ldots, S\}$ is the token position in a sequence of length S and $i \in \{1, \ldots, H\}$ is the dimension.

2.5.6 Residual connections

Residual connections or skip connections (HE *et al.*, 2015) is a technique developed to allow gradients to flow through a network directly, without going through non-linear activation functions in feed-forward layers.

2.5.7 LayerNorm

Layer normalization or LayerNorm (BA *et al.*, 2016) is a technique applied to normalize the distributions of intermediate layers in neural networks. It works by estimating statistics for each example in a batch to allow faster training and better generalization accuracy.

2.5.8 Encoder

We refer to an individual encoder layer as an encoder and use encoder stack for a group of encoder layers. All encoders are identical to each another.

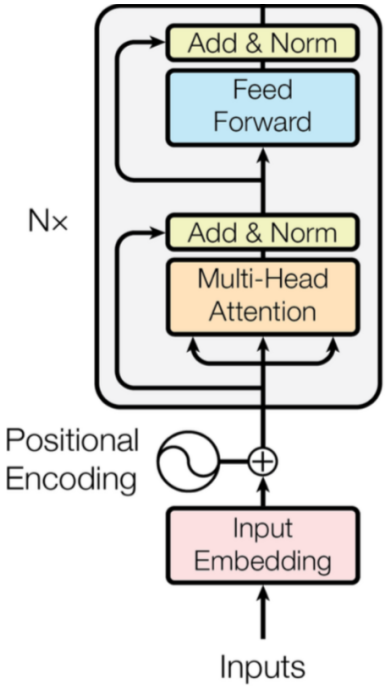

Figure 6 – Encoder architecture. Figure from "Attention Is All You Need" by Vaswani et al. (2017).

The encoding component is a stack of six encoders on top of each other. As shown in Figure 6, each encoder has its own set of weights and consists of two main components: a self-attention mechanism, which calculates the relationship between different words in the sequence, as well as a feed-forward layer, which processes each self-attention output. The encoder also contains the residual skip connections along with the LayerNorm layers (BA *et al.*, 2016).

The first encoder receives the embeddings and positional information of the sequence as input, and the other encoders on the stack receive their input from the previous encoder. Positional information is important for using sequence order, as the transformer processes inputs in parallel. Encoder inputs first flow through a self-attention layer, then through the LayerNorm layer, and finally fed to a feed-forward layer. An architectural detail is that each sublayer in the encoder has a residual connection around it that goes directly to the layer normalization step. The encoder outputs are then passed to the next encoder on the encoder stack as its input, and the output of the last encoder is fed to the first decoder on the decoder stack (VASWANI *et al.*, 2017).

There are many variations of transformer architectures (LIN *et al.*, 2021). Some architectures, such as BERT, do not have a decoder (HE *et al.*, 2020; HE *et al.*, 2021) and rely only on the encoder.

2.5.9 Decoder

Similar to the previous subsection, we refer to an individual decoder layer as a decoder and use decoder stack for a group of decoder layers. Similarly as encoders, all decoders are identical.

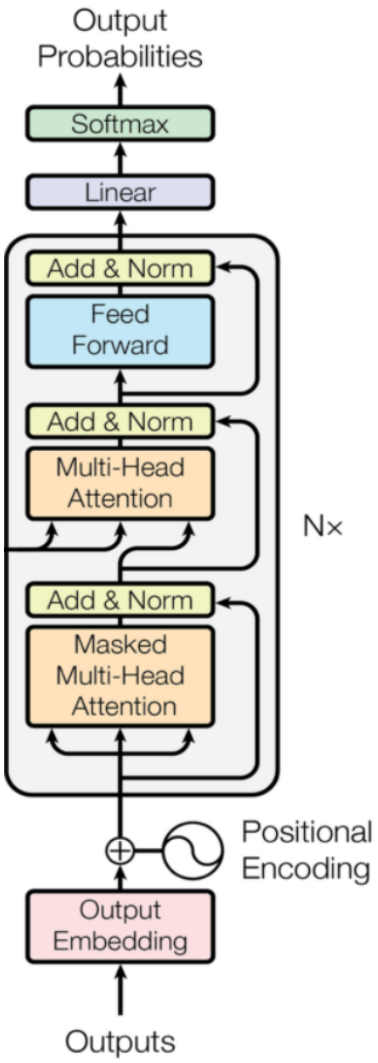

Figure 7 – Decoder architecture. Figure from "Attention Is All You Need" by Vaswani et al. (2017).

The decoder works similarly to the encoder, but as shown in Figure 7, an additional mechanism called the encoder-decoder attention layer is inserted to extract relevant information from the encoders. The encoder-decoder attention layer is located between the self-attention and feed-forward layers and uses the output of the top encoder, transformed into a set of attention arrays, to help the decoder focus on relevant parts of the input sentence. The encoder-decoder attention creates its queries matrix from the layer below it and takes the keys and values matrices from the output of the encoder stack.

The decoding component is a stack of six decoders. Each decoder also has its own set of weights and contains the self-attention layer, the feed-forward layer, residual

skip connections and the LayerNorm layers, as well as an encoder-decoder attention layer. There are important differences between encoder and decoder. Firstly, the output is partially masked, allowing the self-attention layer to only attend to previous positions in the output sequence, and during the decoding process, the output of the decoder from the previous step is fed to the bottom decoder at the current time step. The last decoder is followed by a final linear transformation and a softmax layer to produce the output probabilities over the vocabulary (VASWANI *et al.*, 2017).

2.5.10 The final linear and softmax layers

The output of the decoder stack is an array of floats that go to a linear layer. The linear layer is a feed-forward layer that projects (flattens) the array produced by the decoder stack into a vector of logits, where each element represents a word. Then the softmax layer transforms the logits into probabilities and the word with the highest probability is chosen as the output of the current time step.

2.5.11 Fine-tuning

During fine-tuning, the transformer model learns how to produce a target sequence using the input sequence and the target sequence. In the first step, the input sequence is converted to embeddings and fed into the encoder stack to produce an encoded representation of the input sequence. The target sequence is also converted into embeddings and fed to the decoder to be processed together with the encoded representation of the encoder stack to produce an encoded representation of the target sequence. The output layer converts the encoded representations into word probabilities and generates the final output sequence.

The transformer loss function compares the output sequence with the target sequence of the training data. This loss is used to generate gradients to fine-tune the model using backpropagation (RUMELHART *et al.*, 1986).

2.5.12 Inference

During inference, the transformer model must produce the target sequence from the input sequence only. Similar to the fine-tuning procedure, the input sequence is converted to embeddings and fed into the encoder stack to produce an encoded representation of the input sequence. In the first step, the target sequence is an empty sequence with only a beginning-of-sequence token, which is converted to embeddings and processed by the decoder stack along with the encoded representation of the encoder stack to produce an encoded representation of the target sequence. The output layer converts the encoded representations into word probabilities that are used to generate a predicted word. As

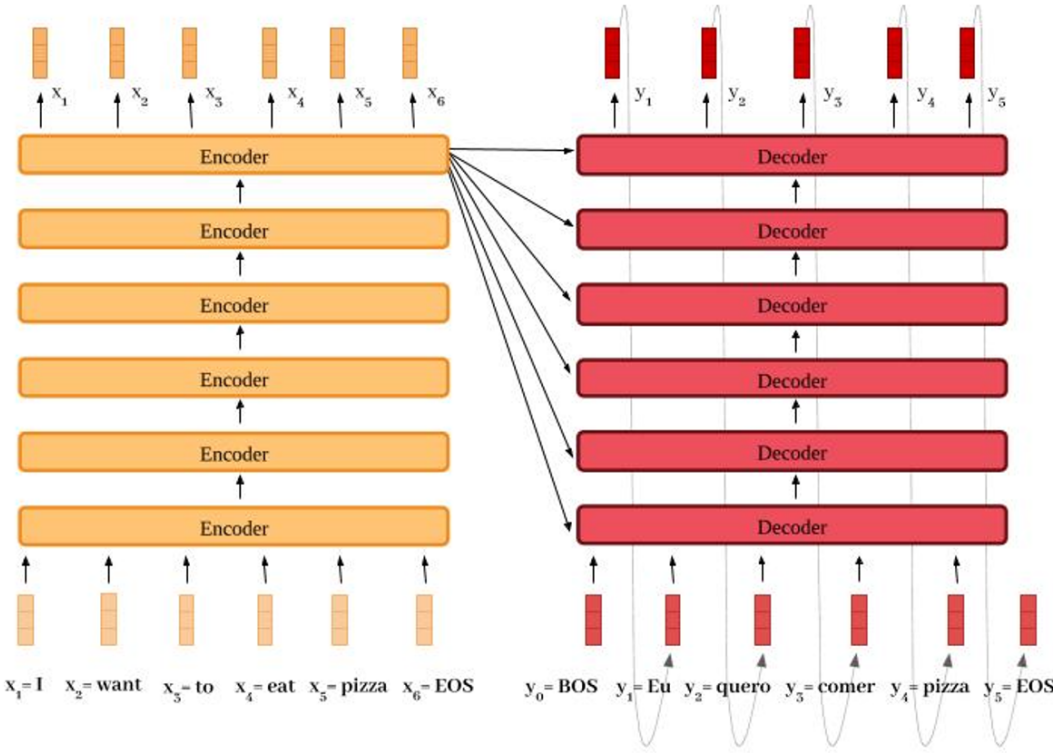

Figure 8 – Transformer inference.

shown in Figure 8, the model generates the output sequence in a loop and feeds the predicted words from previous time steps to the decoder in the next time step until it predicts an end-of-sequence token.

2.6 Transfer learning

Transfer learning consists of training a model on a specific source task and then using the learned weights to start the fine-tuning process on a second task of interest, which is commonly called a downstream task. It is an effective learning technique to reduce the amount of data and training requirements needed to achieve high performance on multiple tasks using neural networks (FARAHANI *et al.*, 2021). This is especially useful when we do not have enough labeled data to fine-tune a good model, because transfer learning allows to get around this problem and develop skillful models that would be unfeasible in the absence of transfer learning.

The main objective is to transfer as much knowledge as possible to develop models capable of leveraging their previously learned representations, which can be useful for performing well in a wide range of tasks (SAGEL *et al.*, 2020). Transfer learning tends to work if the features learned by the model in the first task are general, that is, suitable for both the source and target scenarios, rather than specific to the source only. Basically, the model must be able to apply the knowledge learned in a source scenario to solve a task

in a related target scenario. Choosing data for the source scenario is an open problem in the field and requires domain expertise.

One of the best known and most used types of transfer learning is the pretraining procedure applied in the areas of natural language processing and computer vision, where models are often pretrained in language modeling and image classification tasks, respectively. The datasets used for pretraining are much larger than the datasets for downstream tasks. For example, the ImageNet dataset (DENG *et al.*, 2009) used in computer vision contains 1 million labeled images for classification into 1000 classes, while the Wikipedia and Book corpus dataset, used for BERT pretraining (DEVLIN *et al.*, 2018), consists of 11,038 unpublished books from 16 different genres and 2,500 million words of text passages from Wikipedia in English. For natural language processing, a big advantage of using transfer learning is that we can use a self-supervised approach, in which training examples are automatically generated from raw textual data readily available in most languages, as opposed to supervised datasets that require manually labeled examples. Using a pretrained model, we can achieve high performances with little training data using transfer learning. This is especially useful because training a deep neural network from scratch demands huge amounts of labeled data and can take weeks.

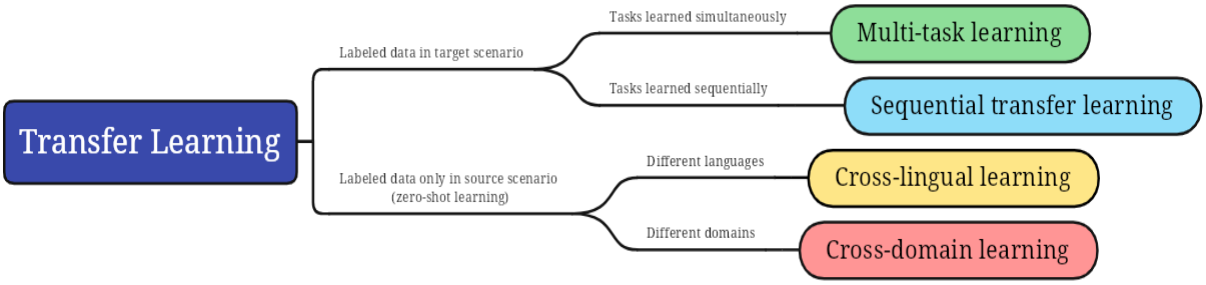

Figure 9 – Transfer learning categories. This dissertation focuses on cross-lingual and cross-domain learning. Figure adapted from Ruder (2019).

Transfer learning techniques have been extensively studied in many machine learning applications. For example, Zhuang et al. (2020) reviewed more than 40 representative transfer learning approaches in an attempt to summarize existing studies and systematize transfer learning strategies for a better understanding of the research field. Their results demonstrate the importance of selecting appropriate transfer learning strategies for different applications and that new approaches are needed to solve knowledge transfer problems in more complex situations, such as real-world scenarios. Pan et al. (2010) also categorized and reviewed progress in knowledge transfer approaches and other related techniques such as domain adaptation and multi-task learning. They also identified and explored some potential future issues in transfer learning research, such as defining a criterion for measuring similarity across domains or tasks and transferring knowledge between unrelated source and target domains. In his doctoral thesis, Ruder

(2019) adapts the taxonomy of Pan et al. to contemporary NLP research. As shown in Figure 9, he categorized transfer learning for NLP into four areas: domain adaptation, cross-lingual learning, multi-task learning and sequential transfer learning. We delve into the areas of cross-lingual learning and domain adaptation, which we refer as cross-domain. In this dissertation, we mainly focus on knowledge transfer between different languages and text domains in natural language processing. In the next sections, we talk about zero-shot learning and provide an overview of both categories of transfer learning that we study throughout this work.

2.7 Zero-shot learning

In many real-world scenarios, annotated data is scarce or even totally unavailable. To handle this situation, we need to take transfer learning to the extreme and use a zero-shot learning approach. The objective is to develop models that perform well in a downstream task without having been trained on a single labeled example (zero-shot learning), or very few examples (few-shot learning) of that task (BENDRE *et al.*, 2020). Success depends on careful selection of the source dataset, but if done correctly, these models can achieve high results.

In a zero-shot approach, during inference, a model receives samples from a data distribution that is different from the samples observed during fine-tuning, or the model has to map its output to a different feature space (e.g., different classes between training and test data). Because machine learning models learn a function from the data distribution seen during fine-tuning, the model must be able to generalize to unseen data at test time. The main goal is to transfer knowledge that may come from related languages, tasks or domains to a target scenario whose specifications are unknown to the model during fine-tuning (POURPANAH *et al.*, 2020). Thus, how well the model is at knowledge transfer plays a crucial role in zero-shot learning.

Furthermore, zero-shot and few-shot models are becoming more competitive with fine-tuned models (BROWN *et al.*, 2020). This ability has been sparking a growing interest in these methods, which today are an active area of research (SCHICK; SCHÜTZE, 2020; LU *et al.*, 2020; TAM *et al.*, 2021).

2.8 Cross-lingual learning

A promising approach to extending the benefits brought by NLP-based technologies to as many languages as possible is to use a single model that can handle multiple languages simultaneously, that is, a multilingual model. Some examples of multilingual models include mBERT (DEVLIN *et al.*, 2018), XLM (CONNEAU; LAMPLE, 2019) and

XLM-R (CONNEAU *et al.*, 2020). These models can handle up to 100 languages and have some common features such as shared dictionary across languages and cross-lingual embeddings, but also their particularities. For example, XLM uses parallel data in its pretraining objective in an attempt to help the model learn similar representations in different languages, while XLM-R is pretrained on a non-parallel multilingual dataset. The results showed that multilingual models are competitive and may even perform better than monolingual models, especially for low-resource languages.

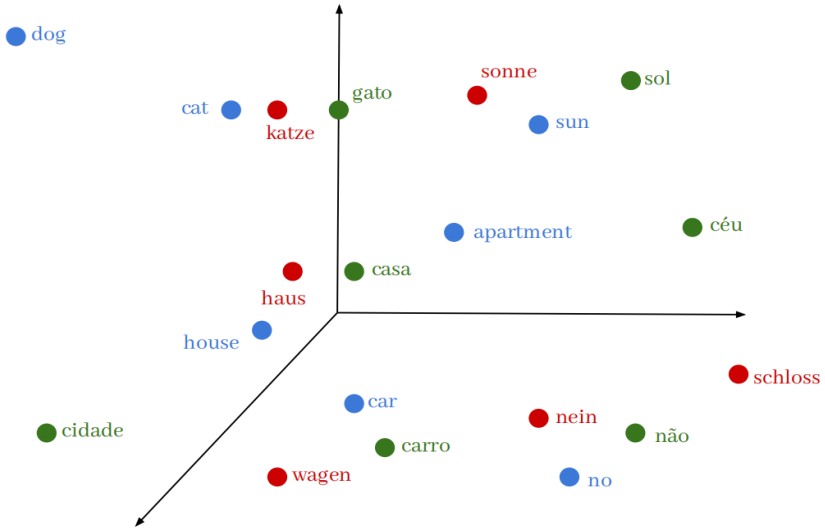

Figure 10 – Cross-lingual word embeddings mapped to the same vector space. Words with similar meaning but in different languages should be located closer.

The need to transfer knowledge across languages has given rise to new features, such as cross-lingual word embeddings (RUDER *et al.*, 2017), which have received increasing attention due to their applicability in multilingual models. Cross-lingual word embeddings allow us to represent and compare the meaning of words in multilingual contexts, facilitating knowledge transfer across languages. This is possible because, as shown in Figure 10, cross-lingual word embeddings provide a joint representation of words from different languages in the same vector space, learning a mapping from monolingual embeddings where words with similar meaning but in different languages should be located closer. This is particularly useful for transferring knowledge from high-resource languages to low-resource languages.

One approach to learn cross-lingual word representations is from parallel data (sentences or documents). Parallel multilingual data capture valuable linguistic information that can be applied in many scenarios. The best known is machine translation, where NLP models can be fine-tuned on large collections of text and their respective human translations (STEINBERGER *et al.*, 2006). Furthermore, parallel datasets can be useful for developing resources for low-resource languages.

A future research direction for cross-lingual learning includes gaining a better

understanding of patterns across language groups. Exploring these patterns can allow the development of models that generalize better to related languages. Another possible direction is to build larger models capable of handling more languages in their parameters. Extending the cross-lingual benchmarks to a wide range of languages and tasks will also be important to assess the knowledge of our models. Cross-lingual learning will be covered in the next chapter.

2.9 Cross-domain learning

In less realistic scenarios, training and test data are usually assumed to come from the same probability distribution. Furthermore, it is widely known that a model fine-tuned on enough labeled examples can achieve high performance when inferring from similar examples. However, when models are applied to real-world scenarios, this assumption often fails, as real-world data can be more diverse and differ from the data seen during fine-tuning (FARAHANI *et al.*, 2020).

Cross-domain learning is often necessary when applying NLP models to real-world scenarios as it deals with the challenge of transferring knowledge from a source domain data distribution to a different target data distribution at test time. This is achieved by learning representations that are useful to a target domain from data coming from a source domain. Cross-domain strategies can be applied to situations where one has a sufficient number of labeled examples in the source domain and few or no labeled examples in the target domain. The goal is to develop more robust models that are able to identify common features across domains to transfer knowledge and generalize to examples outside the source data distribution (DASH *et al.*, 2021). Some examples of adapting to a new domain include a model trained on general domain data adapted to handle data from the legal or biomedical domain, or a model trained to diagnose older viral diseases adapted to detect a new disease such as COVID-19.

Analogous to cross-lingual learning, a possible research direction is to develop models with greater capacity to build common representation spaces in which different domains are closer. Also extending test sets to out-of-domain examples will be crucial for measuring the robustness of our models and assessing how well they can generalize to new domains.

2.10 Conclusions

In this chapter, we have introduced a background knowledge of deep learning and natural language processing needed for subsequent chapters. We also have presented the transformer architecture in detail, as all the NLP models we work with throughout

this dissertation are based on this same architecture. We also have discussed transfer learning, a useful learning strategy that improves the generalization ability of deep learning models and deals with the transfer of knowledge across different NLP tasks, languages and text domains. In addition, we have introduced zero-shot learning and then cross-lingual and cross-domain learning, the two categories of transfer learning that we study in this dissertation.

3 Cross-lingual transfer learning

This chapter addresses the challenge of cross-lingual transfer learning and emphasizes the need to expand resources, especially models and datasets, to enable the development of high-performance models for the huge variety of languages spoken around the world. In many languages, a common problem when using deep learning models for natural language processing tasks is the low availability of high-quality datasets for fine-tuning (HUANG *et al.*, 2019), as data annotation is costly both in terms of money and time spent (DANDAPAT *et al.*, 2009; SABOU *et al.*, 2012). In contrast, recent work shows that multilingual pretrained models achieve surprisingly good cross-lingual zero-shot performance, i.e., models fine-tuned only on a dataset of a high-resource language, such as English, perform well in another language, but on the same task (WU; DREDZE, 2019; PHANG *et al.*, 2020; CONNEAU; LAMPLE, 2019; XUE *et al.*, 2021). This zero-shot cross-lingual ability allows one to use these models on tasks in languages in which annotated data is rare.

These results have sparked a growing interest from the scientific community in extending the advances made in NLP to an increasing number of languages, and consequently, multilingual models are now able to achieve comparable performance to their monolingual counterparts. Notable examples of such models are mBERT (DEVLIN *et al.*, 2018), XLM (CONNEAU; LAMPLE, 2019) and XLM-R (CONNEAU *et al.*, 2020). Their effectiveness is quite surprising, since they are not generally pretrained on any cross-lingual objective. This behavior fostered several studies that aimed to understand and explore it. For example, Wu et al. (2019) explored the cross-lingual potential of multilingual BERT (mBERT) as a zero-shot language transfer model for NLP tasks such as named-entity recognition (NER) and parsing. They further observed that mBERT performs better in languages that share many subwords. Pires et al. (2019) have shown that mBERT has good zero-shot cross-lingual transfer performance on NER and POS tagging tasks.

Artetxe et al. (2019) concluded that neither shared subwords vocabulary nor joint training across multiple languages are necessary to obtain cross-lingual capabilities. They have shown that monolingual models are also capable of performing cross-lingual transfer. K et al. (2019) used mBERT to study the impact of linguistic properties of languages, the architecture of the model, and the learning objectives on the generalization ability of cross-lingual language models. The experiments were conducted in three typologically different languages and they concluded that the lexical overlap between languages contributes little to the cross-lingual success, while the depth of the network plays an important role.

Moreover, due to improvements in machine translation in the last few years (WU *et al.*, 2016; LEPIKHIN *et al.*, 2020), automatically translating datasets from a high-

resource to a low-resource language has also become an effective cross-lingual transfer strategy. Conneau et al. (2019) proposed two methods to pretrain cross-lingual language models, one unsupervised and the other supervised that achieved state-of-the-art results on cross-lingual classification, unsupervised and supervised machine translation. They also have shown that cross-lingual language models can provide significant improvements on the perplexity of low-resource languages. Conneau et al. (2020) presented a transformer-based multilingual model named XLM-R, pretrained on one hundred languages and a strong competitor to monolingual models on several zero-shot benchmarks. Shi et al. (2020) showed that multilingual models perform well on cross-lingual document ranking tasks. They also investigated translating the training data and the translation of documents at inference time and concluded that both approaches achieve competitive results.

Cabezudo et al. (2020) analyzed multiple approaches of using mBERT for the task of natural language inference in Portuguese. They investigated the consequences of adding external data to improve training in two different forms: multilingual data and an automatically translated corpus. They achieved the state of the art on ASSIN corpus using a multilingual pretrained BERT model and showed that using external data did not improve model performance or the improvements are not significant. Rodrigues et al. (2020) showed that automatically translating examples from Portuguese to English and using a model fine-tuned on an English dataset can outperform multilingual models fine-tuned directly on a Portuguese dataset. Isbister et al. (2021) demonstrated that a combination of English language models and modern machine translation outperforms native language models in most Scandinavian languages on sentiment analysis. They argued that it is more effective to translate data from low-resource languages into English than to pretrain a new language model on a low-resource language. Our study expands this work to the tasks of question answering, natural language inference and passage text ranking. Both Xue et al. (2021) and Goyal et al. (2021) compared zero-shot and translation of training data approaches, achieving the best results with the latter.

However, the current literature mainly focuses on developing and understanding transfer learning methods that potentially lead to a better model with respect to some performance metrics (e.g. accuracy or F1 score), ignoring development costs, such as translation of training data, and recurring costs, such as inference cost per example. We extend previous investigations by quantifying the relevant costs embedded during the development and deployment of cross-lingual models. This study analyzes the feasibility and cost-effectiveness of cross-lingual methods to answer the following research question: given the availability of large supervised datasets in English and models pretrained on various languages, what is the most cost-effective way to use these resources for tasks in other languages? The answer to this question will allow us to effectively develop and deploy natural language processing systems for tasks where there is not sufficient labeled data to fine-tune the models. To answer it, we analyze the following transfer learning

methods: 1) fine-tuning a model on a source language and evaluating it on the target language without translation, i.e., in a zero-shot manner; 2) automatic translation of the training dataset to the target language; 3) automatic translation of the test set to the source language at inference time and evaluation of a model fine-tuned in English.

Our main contribution is to evaluate cross-lingual transfer learning methods while also considering their financial and computational costs. In addition, we compare two different translation approaches and show that automatically translating question answering datasets is not trivial. To deal with this, we propose a new method for translating question answering datasets. Furthermore, our cross-lingual models achieve competitive results and can even outperform models fine-tuned directly in the target language, suggesting that in scenarios of lack of resources in the target language, a cross-lingual method can be a great solution. Finally, while exploring the best cross-lingual methods, we reached the state of the art in two datasets used in this study, thus showing that our methodology is sound.

3.1 Models

In this section, we explain the models used in this work. We use BERT, BERTimbau, mBERT, BM25 and mT5 models for the cross-lingual study.

BERT-en: Bidirectional Encoder Representations from Transformers (BERT) is an open-sourced pretrained language model based on the transformer architecture (VASWANI *et al.*, 2017) and a major breakthrough in NLP that quickly achieved state-of-the-art results for a wide variety of tasks. BERT differs from previous language models primarily because of two technical innovations: its learned representation for a specific word contains context from both sides of the sentence, and the model can be pretrained using the huge piles of unannotated data available on the internet, helping to address the problem of low availability of labeled data by drastically reducing the amount of samples needed to learn new tasks. The BERT architecture comprises only the encoder part of the transformer and the model is available in two sizes (Base and Large).

BERT-pt: BERTimbau (SOUZA *et al.*, 2020) is a BERT model pretrained on brWaC (FILHO *et al.*, 2018), a Brazilian Portuguese corpus containing 2.68 billion tokens from 3.53 million documents. BERTimbau improved the state of the art in tasks such semantic textual similarity, natural language inference and named entity recognition in Portuguese.

mBERT: mBERT (DEVLIN *et al.*, 2018) is a BERT model pretrained using the Masked Language Model (MLM) objective on Wikipedia articles in 104 languages with a shared word piece vocabulary. The model follows the same architecture of BERT.

BM25: BM25 is a bag-of-words retrieval function that scores a document based on the query terms appearing in it. We use BM25 implemented in Pyserini (LIN *et al.*, 2021), a Python toolkit that supports replicable information retrieval research.

T5: Raffel et al. (2019) introduced the "Text-To-Text Transfer Transformer" model or T5. This model presents a unified structure that casts language tasks into a text-to-text framework, in which inputs and outputs are texts. This format provides a simple way to perform various tasks, such as machine translation, summarization, question answering, and classification, using the same model architecture, loss function and decoding procedure. T5 is based on the Transformer model originally proposed by Vaswani et al. (2017) with minor differences such as activation function and positional embeddings. The model is pretrained on the Colossal Clean Crawled Corpus (C4), a preprocessed version of publicly available texts extracted from the web. The model achieved state-of-the-art performance in several natural language processing tasks and is provided in five different sizes (Base, Small, Large, 3B and 11B).

mT5: Multilingual T5 or mT5 (XUE *et al.*, 2021) is a multilingual variant of T5 that was pretrained on the mC4 dataset, which contains 101 languages. The model architecture and training procedure is based on T5 with small differences, i.e., the increase in the number of parameters that comes from a larger vocabulary to handle all languages.

3.2 Tasks

In this section, we present the tasks used to evaluate our models.

3.2.1 Question answering

Question answering (QA) is a key problem in the field of NLP that seeks to develop models capable of reading texts and then answering questions about what has been read. It is a task that can be seen as quite challenging, as it requires understanding of natural language and also knowledge about the world.

The answers to the questions can be classified between extractive answers, descriptive answers and multiple choice answers according to their respective formats. Some examples of datasets include XQuAD (ARTETXE *et al.*, 2019), NewsQA (TRISCHLER *et al.*, 2017) and SQuAD (RAJPURKAR *et al.*, 2016). Typically, the datasets for this task are composed of documents and questions to test the model's ability to understand.

For evaluating question answering datasets, the primary metric is the token-level F1, i.e., prediction and ground truth answers are treated as bags of tokens and then their F1 score is calculated. This metric measures the average overlap between prediction and ground truth answers.

$$\mathbf{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{3.1}$$

where *Precision* is the number of common words between the ground truth and the predicted answer divided by the number of predicted words, and *Recall* is the number of common words between the ground truth and the predicted answer divided by the number of words in the ground truth.

3.2.2 Natural language inference

Natural language inference (NLI) is the task of determining whether the meaning of one sentence can probably be inferred from another. More formally, a sentence entails another sentence if, in all cases where the interpretation of the first sentence is true, the interpretation of the second sentence is also true. Textual entailment measures the understanding of natural language because it requires a semantic interpretation of the text. It is an important prerequisite in many NLP applications, such as question answering, information extraction, summarization and neural machine translation, where it is necessary for the model to be able to understand different types of input text and, from that, infer the desired output. Some examples of NLI datasets include MNLI(WILLIAMS *et al.*, 2018), XNLI (CONNEAU *et al.*, 2019) and ASSIN (FONSECA *et al.*, 2016).

For natural language inference datasets, we use classification accuracy as our metric. Classification accuracy summarizes the performance of a model as the number of correct predictions divided by the total number of predictions. It is one of the most common metrics used to evaluate models because it is easy to calculate and intuitive to understand.

$$\mathbf{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives respectively.

3.2.3 Passage ranking

The goal of passage ranking is to generate an ordered list of retrieved texts according to their relevance to a specific query that maximizes some ranking metric (i.e., nDCG or MRR@10). The simplest formulation is to implement a classifier that estimates the probability of each document to belong to the "relevant" class and then ranks all candidates according to these probabilities. There are several datasets used to evaluate the effectiveness of retrieval and ranking models, such as MS MARCO (BAJAJ *et al.*, 2018) and TREC Robust04 (VOORHEES, 2004).

For passage ranking datasets, we use the mean reciprocal rank of the top 10 passages (MRR@10) as it is a measure to evaluate systems that return a ranked list of answers to a specific query. For multiple queries Q, the MRR is the average of the Q reciprocal ranks.

$$\mathbf{MRR} = \frac{1}{Q}\sum_{i}\frac{1}{rank_i} \tag{3.3}$$

3.3 Datasets

This section describes the datasets used throughout the experiments.

3.3.1 SQuAD

The Stanford Question Answering Dataset (SQuAD) (RAJPURKAR *et al.*, 2016) is a question answering dataset whose objective is: given a question and a context, to find the answer as a context span. To set up SQuAD, the authors sampled 536 of the top 10,000 Wikipedia articles. From each of these articles, a total of 23,215 individual paragraphs were extracted and, for each selected paragraph, a group of people was asked to formulate and answer a maximum of five questions about its content. The dataset consists of 107,785 question, context and answer triples manually annotated. From this, 80% of the examples were destined for the training set, 10% for the development set and 10% for the test set.

3.3.2 FaQuAD

FaQuAD (SAYAMA *et al.*, 2019) is a question answering dataset in Portuguese, whose domain is a collection of documents about Brazilian higher education institutions. The dataset follows the SQuAD format and consists of 837 questions for training and 63 for testing, covering 249 paragraphs taken from 18 official documents from a computer science faculty of a Brazilian federal university and 21 Wikipedia articles related to the Brazilian higher education. Text fragments were presented to human annotators and asked to answer questions from each. Each question has between one and three answers and relates to a section of text, usually a paragraph in a document, that includes the correct answer to the question. The objective is the same as SQuAD, i.e., to predict an answer span given a question and a context as input.

3.3.3 GermanQuAD

GermanQuAD (MOLLER *et al.*, 2021) is a mannually annotated dataset inspired by existing QA datasets, such as SQuAD and Natural Questions (KWIATKOWSK *et al.*, 2019). The dataset consists of 13,722 extractive question/answer pairs taken from German

Wikipedia. The training set has 11,518 examples, while the test set has 2,204 examples and they do not overlap. In addition, they also include complex questions that cannot be answered with only a few words. GermanQuAD follows SQuAD in data format and objective.

3.3.4 ViQuAD

The Vietnamese Question Answering Dataset (NGUYEN *et al.*, 2020b) is a dataset for evaluating machine reading comprehension models. The dataset consists of 23,074 human-generated question-answer pairs based on 5,109 passages of 174 Vietnamese Wikipedia articles, in which 18,579 examples are used for training, 2,285 for development, and 2,210 for testing. ViQuAD also follows SQuAD in data format and objective.

3.3.5 MNLI

The Multi-Genre Natural Language Inference (MNLI) (WILLIAMS *et al.*, 2018) is a collection of nearly 433,000 sentence pairs with annotations of entailments, divided into 392,702 examples for training, 20,000 for evaluation and 20,000 for testing. It is designed for use in the development of machine learning models for text comprehension, making possible to perform large-scale natural language inference that seeks to capture the complexity of the English language. Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis, contradicts the hypothesis or neither. The corpus consists of sentences derived from ten different sources, reflecting ten different genres of written and spoken English, which include transcribed speech, letters, fiction and government reports.

The methodology applied to build the MNLI consisted of selecting a sentence from one of the ten textual genres that serves as a premise and asking a human annotator to compose three new sentences in which one must be true, one necessarily false and a neutral sentence. This method of data collection ensures that each class will be represented equally in the final corpus.

3.3.6 XNLI

XNLI (CONNEAU *et al.*, 2019) is an evaluation set for cross-lingual natural language inference created by translating the MNLI development and test sets to 15 languages, including German and Vietnamese. The dataset consists of 5,000 test and 2,500 development examples per language.

3.3.7 ASSIN2

Avaliação de Similaridade Semântica e Inferência Textual (ASSIN2) (REAL *et al.*, 2020) is the second edition of ASSIN, a shared task in Portuguese that evaluates two types of relationship between sentences: semantic textual similarity, which consists of quantifying the level of semantic equivalence between two sentences, and textual entailment recognition (which in this study we refer as natural language inference), whose objective is to classify whether a first sentence entails a second one. The ASSIN2 dataset consists of 10,000 pairs of sentences, of which 6,500 are used for training, 500 for validation, and 2,448 for testing. All pairs were annotated by at least four native speakers of Brazilian Portuguese with linguistic training and only pairs annotated in the same way by most annotators were included in the dataset. ASSIN2 is an effort to offer the community a new benchmark in Natural Language Processing in Portuguese.

3.3.8 MS MARCO

MS MARCO passage ranking (BAJAJ *et al.*, 2018) is a large-scale dataset comprising 8.8M passages taken from the top 10 results retrieved from the Bing search engine using 1 million queries. The objective is to generate an ordered list of retrieved texts according to their relevance to a specific query that maximizes some ranking metric. The training set consists of 530,000 query-passage pairs. The development and test sets consist of 6,900 queries each. Test set annotations are kept hidden and a public submission to the leaderboard is required to assess the effectiveness of the model.

3.3.9 mMARCO

mMARCO (BONIFACIO *et al.*, 2021) is a multilingual dataset created from MS MARCO translated into 8 different languages. All dataset features, such as the number of passages, queries, and relevant query-passage pairs were preserved during translation.

3.4 Experiments

In this section, we explain in detail our experimental setup. First, we describe the three cross-lingual methods. Then we explain the translation procedures and how the costs are calculated. Finally, we describe the training and inference steps.

3.4.1 Cross-lingual methods

Below we describe the methods explored in this study for transferring knowledge in data and models from a source language to a target language.

3.4.1.1 Zero-shot

Zero-shot cross-lingual transfer refers to the strategy of transferring knowledge learned through datasets and models available in a source language, in which ample resources are usually available, to perform tasks in a target language, which typically has fewer labeled data. An example is fine-tuning mBERT on an English dataset, such as SQuAD, and evaluating it directly on a question answering dataset in another language, such as Portuguese, German or Vietnamese.

3.4.1.2 Translate-train

Typically, a high-quality NLP system is fine-tuned on a large dataset. However, many of such datasets are only available in a few languages, such as English and Chinese. One strategy is to translate these datasets using an automatic translator and fine-tune an NLP model on the translated dataset. The advantage of this method is that the translation only needs to be done once, thus there is no extra cost at inference time. Among the disadvantages are the cost of translating the entire training dataset, artifacts introduced during translation (ARTETXE *et al.*, 2020), and constraints on the input-output format of some tasks, such as extractive QA (more on this in Section 3.4.2).

3.4.1.3 Translate-infer

Due to the existence of several models fine-tuned on high quality English datasets, one strategy is to translate the model's input from the target language to English at inference time. The advantages of this method include the simplicity of implementation and also the availability of high-performance machine translation models and off-the-shelf models fine-tuned on tasks in English. The disadvantages are the possible loss of information due to a noisy translation, the cost of the translation and the longer inference time, as the latency of the translation model will be added to the latency of the whole system.

3.4.2 Dataset translation

In this study, we compare two translation approaches to evaluate our models. In the first approach, which we refer to as *open source*, we use translation models from Tiedemann et al. (2020), which were trained using the Marian-NMT framework (JUNCZYS-DOWMUNT *et al.*, 2018). Marian-NMT is a neural machine translation framework originally written in C++ for fast training and translation. We use the following models available on HuggingFace (WOLF *et al.*, 2019): `Helsinki-NLP/opus-mt-en-ROMANCE` for translating from English to Portuguese, `Helsinki-NLP/opus-mt-en-de` for translating from English to German and `Helsinki-NLP`

`/opus-mt-en-vi` for translating from English to Vietnamese. In the second approach, which we refer to as *commercial*, we use the Google Translate API. This API is a paid translation technology developed by Google that uses NLP models to translate text into over a hundred languages.

As each task has its own input-output format, text size, and annotation style, we use a different translation method for each task, as described as follows.

3.4.2.1 QA datasets

We adapted the translation approach used to create the XQuAD dataset, in which human translators were asked to translate the context while keeping special symbols inserted to mark the answer span (ARTETXE *et al.*, 2019). We translate the context and questions following a similar procedure, but in an automatic manner. To ensure that the final answer is contained within the context, we do not translate the answer separately. Instead, as shown in Figure 11, we mark its beginning and end in the context paragraph using special delimiter symbols (e.g., `<answer_start>` and `<answer_end>`). We then translate the context expecting the model to keep these symbols in the correct positions during translation. Finally, we extract the answer and the position of the answer span from the translated text based on these delimiters.

```
Original
Question: Where is the headquarters of the Congregation of the Holy Cross?
Context: The headquarters of Congregation of Holy Cross are located in <answer_start> Rome <answer_end>.
Answer: Rome

Translated
Question: Onde está a sede da Congregação da Santa Cruz?
Context: A sede da Congregação da Santa Cruz está localizada em <answer_start> Roma <answer_end>.
Answer: Roma
```

Figure 11 – An example of our proposed translation method for extractive QA datasets.

However, this strategy does not always work for open source models. In some examples, at least one of the delimiters was not kept in the translated context. To address this problem, we fine-tune the translation model on examples that include these delimiters. We have noticed that the open source models translate single sentences better than multiple sentences. Thus, we translate each sentence in context independently. Translation using a commercial API is faster and less complicated. We simply translate the entire context at once using the strategy explained above.

Using a variable batch size, equal to the number of sentences in context, translating SQuAD takes about 34 hours using an open source model and 6 hours using a commercial API, while translating GermanQuAD and ViQuAD takes about 23 minutes

each for the open source model and 7 minutes using a commercial API. FaQuAD, due to its small size, takes 1.5 minutes and 15 seconds, respectively.

3.4.2.2 NLI datasets

NLI datasets are much easier to translate as the examples are made up of just two sentences, which are translated independently. Thus, there are no special delimiters to consider. The resulting translations are concatenated to form the translated example. Artetxe et al. (2020) argue that translating the premise and the hypothesis separately could affect the generalization ability of the models, introducing noise in the dataset. Translating the two sentences together would provide more context to the translation model, thus improving the quality of the translation. However, due to the aforementioned difficulties in finding delimiters of a translated text, we did not attempt this approach. Translating the MNLI dataset with an open source model takes about 2 hours and 15 minutes, while using a commercial API takes almost 2 hours, both with a batch size of 32. Meanwhile, translating ASSIN2 and XNLI takes less than 1 minute.

3.4.2.3 Passage ranking datasets

The MS MARCO dataset translation covers eight different languages so far. The translation process was conducted equally for all selected languages and is described in detail in Bonifacio et al. (2021). In this work, we use three different mMARCO languages: German, Portuguese and Vietnamese. We refer to this translation process as the *translate-train* method.

We explore two other translation strategies, named *Strategy 1* and *Strategy 2*, which are inference-time translations. Strategy 1 consists of translating the entire dataset into the source language before reranking and, during inference, translating only the queries. This strategy is preferred for search systems that receive many queries but retrieve from a small collection. The second strategy consists of translating the queries and the top 1000 passages returned by a first stage of a search system (i.e., BM25) before feeding them to the reranker. This strategy is useful when the system retrieves from a large collection, but is expected to process a limited number of queries over its lifetime. Translating 8.8 million passages takes approximately 50 hours using an open source model on a V100 GPU with batch size 32. On the other hand, translating using a commercial API takes about 30 hours.

3.4.3 Translation costs

Here, we explain how the translation costs for the open source models and the commercial API were calculated. We also measure the added latency for translation during inference time.

		SQuAD	**FaQuAD**	**GermanQuAD**	**ViQuAD**
(1)	Number of characters (training)	17,688,764	210,804	3,988,977	4,287,841
(2)	Number of characters (test)	2,068,857	38,250	1,019,461	525,784
(3)	Avg. chars / example (training)	936.11	999.07	346.32	230.79
(4)	Avg. chars / example (test)	1000.89	1006.57	462.55	230.10

Table 1 – Statistics of QA datasets.

		MNLI	**ASSIN2**	**XNLI-de**	**XNLI-vi**
(1)	Number of characters (training)	56,521,137	554,639	-	-
(2)	Number of characters (test)	1,379,958	219,834	778,785	658,487
(3)	Avg. chars / example (training)	144.49	85.33	-	-
(4)	Avg. chars / example (test)	140.87	89.80	155.44	131.43

Table 2 – Statistics of NLI datasets.

	MS MARCO	
	Characters	Chars/example
Collection	3,047,540,622	344.67
Training queries	28,667,746	35.44
Development queries	3,615,835	35.77

Table 3 – MS MARCO statistics.

3.4.3.1 Open source

We use cloud-available GPUs to translate using open source models. As of December 2021, the cost of a V100 GPU on Google Cloud was 2.48 USD per hour and on IBM Cloud was 3.06 USD per hour. We use the average of prices to calculate the cost of translating the training set (one-time cost) and 1,000 test examples (recurring cost) using a batch of 32 examples. Costs for translating the training and test sets are reported in USD and USD per thousand examples.

3.4.3.2 Commercial

We calculate translation costs using the average of Google, IBM and Microsoft translation APIs prices, which at the time of this publication is 20 USD per million characters translated for Google and IBM and 10 USD per million characters translated for Microsoft. We also present in Tables 1, 2 and 3 relevant dataset statistics used to report the costs in Section 3.5.

3.4.3.3 Added latency

We also calculate the added latency for translate-infer, which is the time it takes to translate a single batch using an open source model or using a commercial API. Both are measured in seconds per batch translated.

3.4.4 Training and inference

Below, we describe the training and inference procedures used for each task.

3.4.4.1 QA and NLI

In our QA and NLI experiments, we use BERT-pt, BERT-en and mBERT models from the HuggingFace library (WOLF *et al.*, 2019) fine-tuned on SQuAD and MNLI. Our QA models were fine-tuned on a single Tesla V100 GPU with a constant learning rate of 2e-5, over five epochs, with Adam optimizer (KINGMA; BA, 2014), batch size of 12 and a maximum of 384 input tokens. Our NLI models were fine-tuned with a V100, learning rate of 2e-5 over three epochs, batch size of 32 and using Adam optimizer with an input of 128 tokens.

We experiment with two variants of each training set to fine-tune our BERT models: the originals in English and the machine-translated versions in Portuguese, German and Vietnamese. We do the same to evaluate each test set, but, in this case, the originals are in Portuguese, German and Vietnamese and the machine-translated versions are in English. In summary, we try four different strategies: 1) Fine-tune BERT-pt or mBERT in English and evaluate in the target language (zero-shot); 2) Fine-tune BERT-pt or mBERT on a target-translated training set and evaluate it in the same target language (translate-train); 3) Fine-tune BERT-en on SQuAD or MNLI and evaluate it on the test set translated to English (translate-infer BERT-en); 4) As a control experiment, fine-tune BERT-pt or mBERT in English and evaluate it on the test set translated to English (translate-infer BERT-target).

3.4.4.2 Passage ranking

We fine-tune mT5 on the passage ranking task following a similar procedure proposed by Nogueira et al. (2020). The input sequence is formed by a query-passage pair and the model is fine-tuned to return the tokens “yes” or “no”, for German, Portuguese and Vietnamese languages.

All of our fine-tuned models are evaluated on the development set of a translated version of MS MARCO. Additionally, to explore the mT5’s zero-shot performance, we fine-tune the model on MS MARCO dataset in English and evaluate it in the three target languages used in this study.

As proposed by Xue et al. (2021), we also fine-tune mT5 on a bilingual version of MS MARCO, i.e., a dataset formed by joining the original version of MS MARCO in English with a translated version in Portuguese, German or Vietnamese. In all settings, we fine-tune a mT5-base for 100k steps using batches of size 128, a learning rate of 10^{-3} and the AdaFactor optimizer (SHAZEER; STERN, 2018). We use a Google’s TPU v3-8

to fine-tune and evaluate our models. While fine-tuning takes about 12 hours, inference takes approximately 5 hours. We also established a baseline using BM25 implemented in Anserini (YANG *et al.*, 2017). In addition to providing baseline results, the BM25 also serves as an initial retrieval module for mT5 reranking models. The top 1000 passages ranked according to the BM25 are used as candidate passages for the rerankers.

3.5 Results

Table 4 summarizes our main results considering the performance, costs and latencies involved during the process of fine-tuning and deploying the models.

Method	**Score**			**One-time Cost (USD)**	**Recurring Cost (USD/1k ex.)**	**Added latency (s/batch)**
	PT	DE	VI			
QA		**F1**				
Zero-shot	**0.8240**	**0.6497**	**0.6472**	-	-	-
Translate-train (Open S)	0.7391	0.4602	0.4840	2.77	-	-
Translate-train (Comm)	0.7334	0.6478	0.6109	299.17	-	-
Translate-infer (Open S)	0.6772	0.5868	0.4198	-	0.06	2.50
Translate-infer (Comm)	0.6096	0.5887	0.4200	-	16.78	0.23
NLI		**Accuracy**				
Zero-shot	0.8291	0.7063	0.7087	-	-	-
Translate-train (Open S)	0.8746	0.7564	0.7489	6.24	-	-
Translate-train (Comm)	**0.8921**	0.7540	**0.7566**	941.67	-	-
Translate-infer (Open S)	0.8628	0.7720	0.7209	-	0.02	0.78
Translate-infer (Comm)	0.8737	**0.7846**	0.7265	-	1.50	0.76
Ranking		**MRR@10**				
Zero-shot	0.2930	0.2796	0.2414	-	-	-
Translate-train (Open S)	0.2850	0.2640	-	141.27	-	-
Translate-train (Comm)	**0.3020**	**0.2917**	**0.2648**	50,793.00	-	-
Translate-infer - Strategy 1 (Open S)		0.3810*		141.27	0.01	0.72[1]
Translate-infer - Strategy 1 (Comm)		0.3810*		50,793.00	0.70	0.72[1]
Translate-infer - Strategy 2 (Open S)		0.3810*		-	16.36	680.64[2]
Translate-infer - Strategy 2 (Comm)		0.3810*		-	5,733	390.86[2]

Table 4 – Main results. The symbol * denotes an upper bound estimate (see text for details). [1]Each batch has 32 queries, and as passages are already translated, only 32 translations are done. [2]Each batch has 32 queries, each paired with 1000 passages, totaling 32k translations.

3.5.1 Question answering task

As explained in Section 3.4.1, we perform experiments using three cross-lingual methods and two different translation approaches, which are shown in Table 5. Zero-shot (row 2) outperforms all other methods in all languages, including achieving at least a competitive performance in two of the three languages (pt and de) when compared to datasets originally created in the target language (row 1a and 1b). It also has the lowest training and inference costs as it does not require any translation.

Training set translation also achieves good results when using a commercial API. However, open source models in German and Vietnamese performed poorly, probably

	Method	Training data	Model	pt	de	vi
(1a)	Souza et al. (2021)	FaQuAD	BERT-pt	0.5928	-	-
(1b)	Souza et al. (2021)	GermanQuAD	BERT-de	-	0.6863	-
(1c)	Souza et al. (2021)	ViQuAD	BERT-vi	-	-	0.7953
(2)	Zero-shot	SQuAD-en	BERT-target	**0.8240**	**0.6497**	**0.6472**
(3)	Translate-train (Open S)	SQuAD-target	BERT-target	0.7391	0.4602	0.4840
(4)	Translate-train (Comm)	SQuAD-target	BERT-target	0.7334	0.6478	0.6109
(5)	Translate-infer (Open S)	SQuAD-en	BERT-target	0.5390	0.6318	0.4356
(6)	Translate-infer (Comm)	SQuAD-en	BERT-target	0.6251	0.5922	0.4439
(7)	Translate-infer (Open S)	SQuAD-en	BERT-en	0.6772	0.5868	0.4198
(8)	Translate-infer (Comm)	SQuAD-en	BERT-en	0.6096	0.5887	0.4200

Table 5 – Results of the QA task. We use the F1-score as our metric for this task.

due to the complexity of the translation procedure. Thus, the performance of translation methods may, in part, be affected by the translation process, since mapping answer spans across languages is not trivial and introduces some errors. Regarding inference-time translation, it is difficult to assert which translation approach is the best. The results are below the other two methods and suggest that the translation from English to other languages has better quality than the reverse.

Considering the computational and financial costs, it becomes even more evident that zero-shot is the best approach to this task. Translating using open source models is cheaper, while using a cloud service generally provides a better translation. But even so, both development times are still much longer than the zero-shot approach.

3.5.2 Natural language inference task

This section analyzes the results of the natural language inference task in the same experiments done before for the QA task. The results are shown in Table 6.

	Method	Training data	Model	pt	de	vi
(1)	Souza et al. (2021)	ASSIN2-pt	BERT-pt	0.8656	-	-
(2)	Zero-shot	MNLI-en	BERT-target	0.8291	0.7063	0.7087
(3)	Translate-train (Open S)	MNLI-target	BERT-target	0.8746	0.7564	0.7489
(4)	Translate-train (Comm)	MNLI-target	BERT-target	**0.8921**	0.7540	**0.7566**
(5)	Translate-infer (Open S)	MNLI-en	BERT-target	0.8068	0.7662	0.7263
(6)	Translate-infer (Comm)	MNLI-en	BERT-target	0.8059	0.7730	0.7281
(7)	Translate-infer (Open S)	MNLI-en	BERT-en	0.8628	0.7720	0.7209
(8)	Translate-infer (Comm)	MNLI-en	BERT-en	0.8737	**0.7846**	0.7265

Table 6 – Results of the NLI task. We use the accuracy as our metric for this task.

BERT fine-tuned on the translated MNLI (rows 3-4) achieves the best performance in two of the three languages and is 2 to 3 points above the BERT-en models fine-tuned on MNLI-en (rows 7-8) and almost up to 5 points above the zero-shot method

(row 2). For German, the best performance is the BERT-en model evaluated on a translated test set (row 8). In this case, it is not so easy to point out the best method considering all our standards of comparison, because in addition to performance, we need to take into account the time and additional cost to translate the training dataset or to translate at inference time.

Fine-tuning BERT on translated MNLI (rows 3-4) provides the best performance metrics in two of the three languages, but as shown in Table 4, translating MNLI can be relatively expensive if using a commercial API. An open source model is considerably cheaper and can provide competitive performances in this task, probably because each example is made up of just two short sentences. The translate-infer method (rows 5-8) has the advantage of using the original training set in English, but requires translation to English of all examples during inference, causing additional latency to the system. On the other hand, the zero-shot method has the advantage of not requiring translation, but its performance is inferior to the two best cross-lingual methods. In summary, choosing the best NLI system is highly dependent on the requirements of the application in which the model will be used and the financial resources available for development and deployment.

Furthermore, all our cross-lingual methods achieve good results compared to the model directly fine-tuned on a dataset originally made in Portuguese (row 1). This shows that in the absence of datasets in the target language, a cross-lingual method is an excellent solution to data scarcity.

3.5.2.1 Combining cross-lingual methods

In addition, we fine-tune BERT-pt models on different datasets composed of combinations of MNLI and ASSIN2. Both datasets were created for the same task, but they have a different number of classes (MNLI has 3 classes and ASSIN2 has 2). Because of this, we combine the datasets in two ways: (1) We fine-tune using all classes, but during evaluation, we remap predictions from class "Contradiction" (only present in MNLI) to class "Neutral"; (2) We remap all MNLI "Contradiction" examples to "Neutral" and fine-tune on both datasets using just two classes. In preliminary experiments, we found that the second method works better, so we only present results for it. The results in Table 7 show that jointly fine-tuning BERT-pt Large on the English MNLI, MNLI translated to Portuguese and ASSIN2 (in Portuguese) results in the state of the art on ASSIN2.

3.5.3 Passage ranking task

The results of the passage ranking task are shown in Table 8. All rerankers improve results compared to our baseline (BM25), thus demonstrating their cross-lingual capability.

Model	Pretrain	Fine-tune	F1	Acc
mBERT (SOUZA *et al.*, 2020)	100 languages	ASSIN2-pt	0.8680	0.8680
IPR (RODRIGUES *et al.*, 2019)	100 languages	ASSIN2-pt	0.8760	0.8760
Deep Learning Brasil (RODRIGUES *et al.*, 2020)	EN	ASSIN2-en	0.8830	0.8830
PTT5 (CARMO *et al.*, 2020)	EN & PT	ASSIN2-pt	0.8850	0.8860
BERTimbau Large (SOUZA *et al.*, 2020)	EN & PT	ASSIN2-pt	0.9000	0.9000
BERT-pt Base (ours)	EN & PT	MNLI-en + ASSIN2-pt	0.8990	0.8990
BERT-pt Large (ours)	EN & PT	MNLI-en + ASSIN2-pt	0.9180	0.9179
BERT-pt Large (ours)	EN & PT	MNLI-pt + ASSIN2-pt	0.9195	0.9195
BERT-pt Large (ours)	EN & PT	MNLI-(en+pt) + ASSIN2-pt	**0.9207**	**0.9207**

Table 7 – Test results on ASSIN2 using the official evaluation script.

	Method	Training data	Model	pt	de	vi
(1)	Bonifacio et al. (2021)	-	BM25	0.1410	0.1210	0.1359
(2)	Zero-shot	MS MARCO-en	mT5	0.2930	0.2796	0.2414
(3)	Translate-train (Open S)	MS MARCO-target	mT5	0.2850	0.2640	-
(4)	Translate-train (Comm)	MS MARCO-target	mT5	0.3020	0.2917	0.2648
(5)	Translate-train (Comm)	MS MARCO-(en+target)	mT5	**0.3060**	**0.2941**	**0.2689**

Table 8 – Results on the development set of Portuguese MS MARCO. We use the MRR@10 as our metric for this task.

Although the zero-shot method is more effective than BM25, we can see in Table 8 that translation improves results when using a commercial API. All models fine-tuned on the API-translated MS MARCO (row 4) perform better than the zero-shot ones (row 2). This suggests that the noise introduced due to translation, in this case, did not harm the learning process. Finally, the models fine-tuned on the original and translated datasets (row 5) outperform all others by a small margin.

MS MARCO is considerably larger than the other datasets used in this work, resulting in a much higher translation cost, as shown in Table 4. However, despite having superior performance when compared to the open source model, translating using an commercial API is almost 360 times more expensive. So, If the collection and the number of queries are large, the zero-shot method is recommended, otherwise, fine-tuning on a bilingual dataset is the best choice.

For translation at inference time (translate-infer), the added latency for strategy 1 is shorter as only queries are translated. On the other hand, the added latency for strategy 2, which translates 1000 passages for each query, is considerably longer. For both strategies of translate-infer, we do not translate back to English the translated versions of MS MARCO, since we already have the original in English. Thus, for these methods, we report the same MRR@10 from Nogueira et al. (2020), who fine-tuned and evaluated a T5 on the original MS MARCO in English. Note that this result is an upper bound estimate as translation would likely introduce artifacts and therefore degrade the quality of the reranker models. To calculate costs, we also estimate that the dataset translated back to English would have similar statistics (e.g., average passage length) to the original.

3.6 Conclusions

In this chapter, we analyzed cross-lingual methods in terms of their effectiveness (e.g., accuracy), development and deployment costs, as well as their latencies at inference time. We conducted experiments on three different tasks and, by combining zero-shot and translation methods, we achieved the state of the art on two datasets used in this work. In the next chapter, we will focus on cross-domain learning methods. We will evaluate and compare our models on a legal domain task.

4 Cross-domain transfer learning

An ongoing trend in natural language processing is to use the same model with minor adaptations to solve a variety of tasks. Pretrained transformers, epitomized by BERT (DEVLIN *et al.*, 2019), are the state of the art in question answering (KHASHABI *et al.*, 2020), natural language inference (HE *et al.*, 2020; RAFFEL *et al.*, 2020), summarization (LEWIS *et al.*, 2020; BAO *et al.*, 2020), and ranking tasks (MA *et al.*, 2021; GAO *et al.*, 2021). Although these tasks are diverse, current top-performing models in each have an architecture similar to the original Transformer (VASWANI *et al.*, 2017) and are pretrained on variations of the masked language modeling objective used by Devlin et al. (2019).

Pretrained transformer models have only begun to be adopted in legal NLP applications more broadly (YEUNG, 2019; ELWANY *et al.*, 2019; SHAGHAGHIAN *et al.*, 2020; LEIVADITI *et al.*, 2020; BAMBROO; AWASTHI, 2021). In some tasks, they marginally outperform classical methods, especially when training data is scarce. For example, Zhong et al., (2020a) showed that a BERT-based model performs better than a tf-idf similarity model on a judgment prediction task (XIAO *et al.*, 2018), but is slightly less effective than an attention-based convolutional neural network (YIN *et al.*, 2016). In some cases, they outperform classical methods, but at the expense of using hand-crafted features or by being fine-tuned on the target task. For example, the best submission to the task of legal case entailment in COLIEE 2019 was a BERT model fed with hand-crafted inputs and fine-tuned on in-domain data (RABELO *et al.*, 2019a). Peters et al. (2019) demonstrate that fine-tuning on the target task may not perform better than simple feature extraction from a pretrained model if the pretraining task and the target task belong to highly different domains. These findings lead us to consider zero-shot approaches as we investigate how general domain transformer models can be applied to legal tasks.

Zero-shot and few-shot models are becoming more competitive with models fine-tuned on large datasets. For example, the GPT-3's excellent few-shot results (BROWN *et al.*, 2020) sparked interest in prompt engineering methods, which are now an active area of research (SCHICK; SCHÜTZE, 2020; LU *et al.*, 2020; TAM *et al.*, 2021). Although zero-shot approaches are relatively novel in the legal domain, our work is not the first to apply zero-shot transformer models to domain-specific entailment tasks when the availability of labeled data is limited. Yin et al. (2019) transformed multi-label classification tasks into textual entailment tasks and then evaluated the performance of a BERT model fine-tuned on mainstream entailment datasets. Yin et al. (2020) also performed similar experiments transforming question answering and coreference resolution tasks into entailment tasks.

We are not the first to use zero-shot techniques on the legal case entailment

task. For instance, Rabelo et al. (2020) used a BERT fine-tuned for paraphrase detection combined with two transformer-based models fine-tuned on a generic text entailment dataset and features generated by a BERT model fine-tuned on the COLIEE dataset. However, we are the first to show that zero-shot models can outperform fine-tuned ones on this task.

It is a common assumption among NLP researchers that models developed using non-legal texts would lead to unsatisfactory performance when applied directly to legal tasks (ZHONG *et al.*, 2020a; ELNAGGAR *et al.*, 2018b). To overcome this problem and enable knowledge transfer across domains, general-purpose techniques were adapted to the legal domain. For example, Chalkidis et al. (2019) pretrained legal word embeddings using word2vec (MIKOLOV *et al.*, 2013; MIKOLOV, 2013) over a large corpus comprised of legislation from multiple countries. Zhong et al. (2020b) created a question answering dataset in the legal domain, collected from the National Judicial Examination of China and evaluated different models on it, including transformers. Elnaggar et al. (2018a) applied multi-task learning to minimize problems related to data scarcity in the legal domain. The models were fine-tuned on translation, summarization, and multi-label classification tasks and achieved better results than single-task models.

Nonetheless, in information retrieval, pretrained models fine-tuned only on a large dataset have shown strong cross-domain zero-shot capabilities (THAKUR *et al.*, 2021). For example, the same multi-stage pipeline based on T5 (RAFFEL *et al.*, 2020) used in this study was either the best or second best-performing system in 4 tracks of the TREC 2021 (PRADEEP *et al.*, 2020), including specialized tasks such as Precision Medicine (ROBERTS *et al.*, 2019) and TREC-COVID (ZHANG *et al.*, 2020). A remarkable feature of this pipeline is that for most tasks the models are fine-tuned only on a general-domain ranking dataset, i.e., they do not use in-domain data.

However, to date, there has not been strong evidence that zero-shot models transfer well to the legal domain. Most state-of-the-art models need adaptations to the target task. For example, the top-performing system on the legal case entailment task of COLIEE 2020 (RABELO *et al.*, 2020) uses an interpolation of BM25 (ROBERTSON *et al.*, 1995) scores and scores from a BERT model fine-tuned on the target task (NGUYEN *et al.*, 2020a).

We show, for the legal case entailment task, that pretrained language models without any fine-tuning on the target task can perform better than models fine-tuned on the task itself. Our approach is characterized as zero-shot since the model was fine-tuned on general domain text and evaluated on legal domain text. Our result confirms, in the legal domain, a recent counter-intuitive finding verified in other domains: given limited labeled data, zero-shot models tend to perform better on held-out datasets than models fine-tuned on the target task (RADFORD *et al.*, 2021; PRADEEP *et al.*, 2020).

The cross-domain study follows a template similar to that used in the cross-lingual study: models, task description and dataset, experiments, and, in the last two sections, the results and conclusions of the chapter.

4.1 Models

In this section, we describe the models used throughout this study. We use BM25 and two transformer-based models, MonoT5 and DeBERTa.

BM25: As explained in Section 3.1, BM25 is a classic search algorithm used to sort documents based on query terms.

monoT5: At a high level, monoT5 is a sequence-to-sequence adaptation of the T5 model (RAFFEL *et al.*, 2020) proposed by (NOGUEIRA *et al.*, 2020) and further detailed in (LIN *et al.*, 2020). MonoT5 is trained on MS MARCO (BAJAJ *et al.*, 2018) and designed to generate the tokens "true" or "false" depending on the relevance of a document to a query. During inference, the model estimates a score that quantifies the relevance of a document to a query. To compute this score for each query-document pair, a softmax is applied only on the logits of the tokens "true" and "false". The final score of each candidate is the probability assigned to the token "true". This ranking model is close to or at the state of the art on retrieval tasks such as Robust04 (VOORHEES, 2004), TREC-COVID, and TREC 2020 Precision Medicine and Deep Learning tracks (PRADEEP *et al.*, 2020).

DeBERTa: *Decoding-enhanced BERT with disentangled attention* (DeBERTa) improves on the original BERT and RoBERTa (LIU *et al.*, 2019) architectures by introducing two techniques: the disentangled attention mechanism and an enhanced mask decoder (HE *et al.*, 2020). Both improvements seek to introduce positional information to the pretraining procedure, both in terms of the absolute position of a token and the relative position between them. The model is the state of the art in many NLP tasks.

4.2 Legal case entailment task

The Competition on Legal Information Extraction/Entailment (COLIEE) (RABELO *et al.*, 2020; RABELO *et al.*, 2019b; KANO *et al.*, 2018; KANO *et al.*, 2017) is an annual competition whose objective is to evaluate automated systems on case and statute law tasks.

Legal case entailment is one of the five tasks of the COLIEE 2021 competition, which consists in identifying paragraphs from candidate cases that entail a fragment of a new legal decision. The purpose of the task is, given a fragment of a new court decision Q, we need to identify a set of paragraphs $P = [P_1, P_2, ..., P_n]$ that are relevant to this new decision.

The micro F1-score is the official metric in this task:

$$\mathbf{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4.1}$$

where *Precision* is the number of paragraphs correctly retrieved for all legal decisions divided by the number of paragraphs retrieved for all legal decisions, and *Recall* is the number of paragraphs correctly retrieved for all legal decisions divided by the number of relevant paragraphs for all legal decisions.

4.3 COLIEE's dataset

The COLIEE 2021 (RABELO *et al.*, 2022) dataset is predominantly composed of Federal Court of Canada case laws. The training data consists of 425 court decision fragments, their respective candidate paragraphs that may or may not be relevant to the fragment, and a set of labels containing the paragraphs by which the decision fragment is entailed. Test data only includes 100 decision fragments and their candidate paragraphs, but no labels. The dataset has an average of 35 candidate paragraphs per court decision fragment, with only one of them being relevant on average. In Table 9, we show the statistics of the 2020 and 2021 datasets.

	2020		**2021**	
	Train	**Test**	**Train**	**Test**
Examples (base cases)	325	100	425	100
Avg. # of candidates / example	35.52	36.72	35.80	35.24
Avg. positive candidates / example	1.15	1.25	1.17	1.17
Avg. of tokens in base cases	37.72	37.03	37.51	32.97
Avg. of tokens in candidates	100.16	112.65	103.14	100.83

Table 9 – COLIEE dataset statistics.

The COLIEE 2020 dataset is the same as the COLIEE 2021 training set. As shown in Figure 12, the input to the model is a decision fragment of an unseen case and the output must be a set of relevant paragraphs.

4.4 Experiments

We experiment with the following models: BM25, monoT5-zero-shot, monoT5, and DeBERTa. We also evaluate an ensemble of our monoT5 and DeBERTa models. First, we explain how we select the final answers of our models.

Input Q: One must remember that this type of application, a citizenship appeal, is not a trial de novo as was once the case, but rather is an appeal by way of an application purely on the record which was before the Citizenship Judge

Candidate 1: [1]The first two criteria are not at issue in this case.
Candidate 2: [2]Under the former Rules, a citizenship appeal was a trial de novo and there was a distinct set of rules governing its conduct. Under the new Rules, citizenship appeals proceed by way of an application based on the record before the citizenship.
Candidate 3: [3]Accordingly the appeal is allowed and the decision of the Citizenship Judge is set aside. Appeal allowed.
Candidate 4: [4]In this case, the Citizenship Court calculated that the respondent was physically present in Canada for 433 days, a shortage of 567 days in the required three year period.

Target P: [2]Under the former Rules, a citizenship appeal was a trial de novo and there was a distinct set of rules governing its conduct. Under the new Rules, citizenship appeals proceed by way of an application based on the record before the citizenship.

Figure 12 – Example of COLIEE dataset.

4.4.1 Answer selection

The models used in the experiments estimate a score for each (fragment, candidate paragraph) pair. To select the final set of paragraphs for a given fragment, we apply three rules:

- Select paragraphs whose scores are above a threshold α;
- Select the top β paragraphs with respect to their scores;
- Select paragraphs whose scores are at least γ of the top score.

We use exhaustive grid search to find the best values for α, β, γ on the development set of the 2020 dataset. We swept $\alpha = [0, 0.1, ..., 0.9]$, $\beta = [1, 2..., 10]$, and $\gamma = [0, 0.1, ..., 0.9, 0.95, 0.99, 0.995, ..., 0.9999]$. The best values for each model can be found in Table 12. Note that our hyperparameter search includes the possibility of not using the first rule or the third rule if $\alpha = 0$ or $\gamma = 0$ are chosen, respectively.

4.4.2 Zero-shot approach

Now, we describe the zero-shot strategy explored in this study for transferring knowledge from a source domain to a target domain.

4.4.2.1 BM25

We use BM25 implemented in Pyserini with its default parameters. The first step in our approach is to index all candidate paragraphs present in the COLIEE 2021 dataset to calculate the term statistics (e.g., document frequencies) used by BM25.

The input to BM25 is a court decision fragment that can be comprised of multiple sentences. Here, we treat each of its sentences as a single query and compute a score for each sentence and candidate paragraph pair independently. The final score for each paragraph is the maximum among its sentence and candidate paragraph pair scores. We then use the method described in Section 4.4.1 to select the paragraphs that will comprise the final answer.

4.4.2.2 monoT5-zero-shot

We use T5 models fine-tuned on MS MARCO, a dataset of approximately 530,000 query and relevant passage pairs. We use checkpoints available in Huggingface's model hub that were fine-tuned with a learning rate of 10^{-3} using batches of 128 examples for 10,000 steps, or approximately one epoch of the MS MARCO dataset.[1] [2] In each batch, a roughly equal number of positive and negative examples are sampled. We refer to these models as monoT5-zero-shot and monoT5-3B-zero-shot respectively.

Although fine-tuning for more epochs leads to better performance on the MS MARCO development set, Nogueira et al. (2020) showed that further training degrades a model's zero-shot performance on other datasets. We observed similar behavior in our task and opted to use the model trained for one epoch on MS MARCO. Our approach is characterized as zero-shot, since the model was fine-tuned on general domain text annotated in MS MARCO and evaluated on the COLIEE dataset, which is composed of legal domain text.

At inference time, for the task of legal case entailment, we use the following input sequence template for all transformer models:

$$\text{Query: } q \quad \text{Document: } d \quad \text{Relevant:} \tag{4.2}$$

where q and d are the query and candidate text, respectively. In this experiment, q is a fragment decision of a new legal case, and d is one of the candidate paragraphs from existing legal cases that may or may not be relevant to a new decision. The model estimates a score s that quantifies how relevant a candidate paragraph d is to a fragment of a court decision q. That is:

$$s = P(\text{Relevant} = 1|d, q). \tag{4.3}$$

The final score of each candidate is the probability assigned by the model to the token "true". After computing all scores, we apply the method described in Section 4.4.1.

4.4.3 Fine-tuning approaches

4.4.3.1 monoT5

We further fine-tune monoT5-zero-shot on the COLIEE 2020 training set following a similar training procedure described in the previous section. The model is fine-tuned with a learning rate of 10^{-3} for 80 steps using batches of size 128, which corresponds to 20 epochs. Each batch has the same number of positive and negative examples. During

[1] https://huggingface.co/castorini/monot5-large-msmarco-10k
[2] https://huggingface.co/castorini/monot5-3B-msmarco-10k

inference, we calculate all scores and apply our answer selection method to generate the set of final answers.

Fragments are mostly comprised of just one sentence, while candidate paragraphs can be longer, sometimes exceeding 512 tokens in length. Thus, to avoid excessive memory usage due to the quadratic memory cost of transformers with respect to the sequence length, we truncate the inputs to 512 tokens during training and inference.

4.4.3.2 DeBERTa

The COLIEE 2020 dataset has very few positive examples of entailment. Therefore, for fine-tuning DeBERTa on this dataset, we found appropriate to artificially expand the positive examples to reduce the imbalance between classes. Since fragments occupy only a small portion of a court decision, we expand positive examples by generating artificial fragments from the same court decision in which the original fragment occurs. This is done by moving a sliding window over the court decision. Each step of this sliding window is considered an artificial fragment, and these artificial fragments receive the same labels as the original fragment.

Although the resulting dataset after these operations is several times larger than the original dataset, we achieved better results by fine-tuning DeBERTa on a small sample taken from this artificial dataset. After experimenting with different sample sizes, we decided on a sample of twenty thousand fragment and candidate paragraph pairs, equally balanced between positive and negative entailment pairs. The model is fine-tuned for ten epochs and the checkpoint with the best performance on the 2020 test set is selected to generate the predictions for the 2021 test set.

4.4.3.3 DebertaT5 (ensemble)

We use the following method to combine the predictions of monoT5 and DeBERTa (both fine-tuned on COLIEE 2020 dataset): We concatenate the final set of paragraphs selected by each model. We remove duplicates, preserving the highest score. Then, we apply again the grid search method explained in Section 4.4.1 to select the final set of paragraphs. It is important to note that our method does not combine scores between models. The final answer for each test example is made up of individual answers from one or both models. It ensures that only answers with a certain degree of confidence are maintained, which generally leads to an increase in precision. Ensemble methods seek to combine the strengths and compensate for the weaknesses of the models to achieve better performance.

4.5 Results

We present our main result in Table 10. Our baseline method (BM25) scores above the median of submissions in both COLIEE 2020 and 2021 datasets (row 2 vs. 1a). This confirms that BM25 is a strong baseline and it is in agreement with results from other competitions such as the Health Misinformation and Precision Medicine track of TREC 2020 (PRADEEP *et al.*, 2020).

		2020			2021			
Description	**Submission name**	**F1**	**Prec**	**Recall**	**F1**	**Prec**	**Recall**	α, β, γ
(1a) Median of submissions	-	0.5718	-	-	0.5860	-	-	-
(1b) Best of 2020 (NGUYEN *et al.*, 2020a)	JNLP.task2.BMWT	0.6753	0.7358	0.6240	-	-	-	-
(1c) 2nd best of 2021	UA_reg_pp	-	-	-	0.6274	-	-	-
(2) BM25	-	0.6046	0.7222	0.5200	0.6009	0.6666	0.5470	0.07, 2, 0.99
(3) DeBERTa	DeBERTa	**0.7094**	0.7614	0.6640	0.6339	0.6635	0.6068	0, 2, 0.999
(4) monoT5	monoT5	0.6887	0.7155	0.660	0.6610	0.6554	0.6666	0, 3, 0.995
(5) monoT5-zero-shot	-	0.6577	0.7400	0.5920	**0.6872**	0.7090	0.6666	0, 3, 0.995
(6) Ensemble of (3) and (4)	DebertaT5	0.7217	0.7904	0.6640	0.6912	0.7500	0.6410	0.6, 2, 0.999
(7) Ensemble of (3) and (5)	-	0.7038	0.7592	0.6560	0.6814	0.7064	0.6581	0.6, 2, 0.999

Table 10 – Test set results on legal case entailment task of COLIEE 2020 and 2021. Our best single model for each year is in bold.

Our pretrained transformer models (rows 3, 4 and 5) score above BM25, the best submission of 2020 (NGUYEN *et al.*, 2020a), and the second best team of 2021. The most interesting comparison is between monoT5 and monoT5-zero-shot (rows 4 and 5). On the 2020 test data, monoT5 showed better results than monoT5-zero-shot. Hence, we decided to submit only the fine-tuned model to the 2021 competition. Following the release of ground-truth annotations of the 2021 test set, our evaluation of monoT5-zero-shot showed that it performs better than monoT5. A similar "inversion" pattern was found for DeBERTa vs. monoT5-zero-shot (rows 3 and 5). DeBERTa is better than monoT5-zero-shot on the 2020 test set, but the opposite happened on the 2021 test set.

As the COLIEE dataset is very small, one explanation for these results is that we may have, unintentionally, overfitted on the COLIEE 2020 data by selecting techniques (e.g., data augmentation) and hyperparameters that gave the best result in these samples as the experiments progressed. This may have caused a generalization difficulty when evaluated on the 2021 competition test set, which was only revealed by the competition organization later. Another explanation is that there is a significant difference between the annotation methodologies of 2020 and 2021. Consequently, models specialized in the 2020 data could suffer from this change. However, this is unlikely since BM25 performs similarly in both years. Furthermore, we cannot confirm this hypothesis as it is difficult to quantify differences in the annotation process.

Regardless of the reason for the inversion, our main finding is that our zero-shot model performs at least comparable to the fine-tuned models on the 2020 test set and achieves the best result of a single model on the 2021 test data, outperforming DeBERTa

Model	**Parameters**	**F1**	**Prec**	**Recall**	α, β, γ	**Description**
DeBERTa	350 millions	0.6339	0.6635	0.6068	0, 2, 0.999	Single model
monoT5	770 millions	0.6610	0.6554	0.6666	0, 3, 0.995	Single model
monoT5-zero-shot	770 millions	0.6872	0.7090	0.6666	0, 3, 0.995	Single model
DebertaT5-zero-shot	(350 + 770) millions	0.6814	0.7064	0.6581	0.6, 2, 0.999	Ensemble
DebertaT5	(350 + 770) millions	0.6912	0.7500	0.6410	0.6, 2, 0.999	Ensemble
monoT5-3B-zero-shot	3 billions	**0.7373**	**0.8000**	**0.6838**	0, 1, 0	Single model

Table 11 – Evaluating zero-shot performance at scale on COLIEE 2021

and monoT5, which were fine-tuned on the COLIEE dataset. To the best of our knowledge, this is the first time that a zero-shot model outperforms fine-tuned models in the task of legal case entailment. Given limited annotated data for fine-tuning and a held-out test data, such as the COLIEE dataset, our results suggest that a zero-shot model fine-tuned on a large general domain dataset may be more robust to changes in data distribution and can generalize better on unseen data than models fine-tuned on a small domain-specific dataset.

Moreover, our ensemble method effectively combines DeBERTa and monoT5 predictions, achieving the best score among all submissions (row 6). However, the performance of monoT5-zero-shot decreases when combined with DeBERTa (row 5 vs. 7), showing that monoT5-zero-shot only is a strong model. It is important to note that despite the performance of DebertaT5 being the best in the COLIEE competition, the ensemble method requires training time, computational resources and perhaps also data augmentation to perform well on the task, while monoT5-zero-shot does not need any adaptation. The model is available online and ready to be used.

4.5.1 Zero-shot capacity at scale

Recent results have demonstrated that language models scaled to billions of parameters, such as GPT-3, perform remarkably well in zero-shot and few-shot scenarios (BROWN *et al.*, 2020; WEI *et al.*, 2021). So, as a last experiment, we evaluate a much larger model on the COLIEE dataset, the monoT5-3B-zero-shot. Our results in Table 11 show that even without using our answer selection approach, monoT5-3B-zero-shot easily outperforms all models, including ensembles, suggesting that stronger zero-shot capability may be a feature of larger models.

4.5.2 Ablation of the answer selection method

In Table 12, we show the ablation result of the answer selection method proposed in Section 4.4.1.

Our baseline answer selection method, which we refer to as “no rule” in the table,

Model	**F1**	**Prec**	**Recall**	α, β, γ
monoT5-zero-shot (no rule)	0.6517	0.7373	0.584	0, 1, 0
monoT5-zero-shot	0.6577	0.7400	0.592	0, 3, 0.995
monoT5 (no rule)	0.6755	0.7600	0.608	0, 1, 0
monoT5	0.6887	0.7155	0.6640	0, 3, 0.995
DeBERTa (no rule)	0.6933	0.7800	0.6240	0, 1, 0
DeBERTa	0.7094	0.7614	0.6640	0, 2, 0.999
DebertaT5-zero-shot (no rule)	0.6875	0.7777	0.6160	0, 1, 0
DebertaT5-zero-shot	0.7038	0.7592	0.6560	0.6, 2, 0.999
DebertaT5 (no rule)	0.7022	0.7900	0.6320	0, 1, 0
DebertaT5	0.7217	0.7904	0.6640	0.6, 2, 0.999

Table 12 – Ablation on the 2020 data of the answer selection method presented in Section 4.4.1.

uses only the paragraph with the highest score as the final answer set, i.e., $\alpha = \gamma = 0$ and $\beta = 1$. For all models, the proposed answer selection method offers improvements of at least 0.6 to 2 F1 points.

4.6 Conclusions

In this chapter, we summarized our participation in the COLIEE competition, described our submissions, and demonstrated that our zero-shot approach is competitive and can outperform models fine-tuned on the target task. In the final chapter, we will look back and highlight the main topics discussed throughout this dissertation, we will also present our conclusions and, at the end, we will provide some insights and reflections on the future of transfer learning in NLP.

5 Conclusion

Throughout this dissertation, we have made contributions to two categories of transfer learning in NLP, which we refer to as cross-lingual and cross-domain. To conclude, in this chapter we will recap the proposed experiments, summarize our results, and provide a discussion about the future of the field.

5.1 Dissertation recap

In this work, we have studied the problem of learning representations that can be successfully transferred to a variety of languages or text domains. In Chapter 2, we have provided an overview of artificial inteligence and its related subfields such as deep learning and natural language processing. We also have presented the transformer architecture and key concepts for this thesis, such as transfer learning, zero-shot learning, cross-lingual learning and cross-domain learning.

Chapter 3 presented cross-lingual learning strategies that seek to bridge differences between languages. We have extensively analyzed three methods for transferring knowledge from a high-resource language to a low-resource language, considering the costs in terms of time and money spent on developing each strategy. In addition, we compared two different translation approaches and evaluated our models on three NLP tasks in three different languages.

Chapter 4 focused on cross-domain learning, particularly on transferring knowledge from general domain texts to domain-specific texts. In this study, we tackled the challenge of texts in the legal domain. We proposed a zero-shot strategy, already successful in other domains, such as the medical domain, and compared it to several models directly fine-tuned in texts in the legal domain.

5.2 Summary of results

This dissertation investigates the hypothesis that, in some scenarios, zero-shot transformer models that leverage useful information using transfer learning can outperform supervised models, especially in low-resource languages and technical text domains, where there is a shortage of resources. Over the course of this thesis, we present strategies for two different categories of transfer learning, explore their feasibility and zero-shot performance, and evaluate them against fine-tuned models across different target tasks. From our experiments, we were able to develop models with great capacity for transferring knowledge between different languages and different domains. Finally, we also confirm our

hypothesis and demonstrate that zero-shot models can outperform fine-tuned models in both the target language and the target domain. Based on our results, we question the common assumption that it is always necessary to label training data in the target language or domain to perform well on a task. Our results suggest that zero-shot approaches are effective and fine-tuning on a large labeled dataset in a high-resource language or in general domain texts may be sufficient for both cases. We now summarize our contributions and findings.

5.2.1 Cross-lingual

In the cross-lingual study, we investigate three methods of transferring knowledge from pretrained transformer models in English to Portuguese, German and Vietnamese in the tasks of question answering, natural language inference and passage ranking, as well as analyzing their development and deployment costs. We compare three alternatives: translating training data, translating test data at the time of inference, and training and inferring without translation, i.e., zero-shot.

Within the methods, we also compare two different translation approaches. The first, which we refer to as open-source, uses translation models from HuggingFace fine-tuned using the Marian-NMT framework. The second, which we refer to as commercial, uses the Google Translate API, a paid neural machine translation technology developed by Google that translates text into over a hundred languages. We show that translating using the Google Translate API is more expensive but faster and generally gives better results than MarianNMT.

Our results reveal that the zero-shot method works well for all tasks, especially for question answering, whose translation process is more complex than the other two tasks. To address this challenge, we propose a new automated method for translating QA datasets inspired by the translation method used to create the XQuAD dataset, in which human translators were asked to translate the context while keeping special symbols inserted around the correct answer. Although the QA translation method worked well for most of the examples, some open-source models found it difficult to keep the special symbols in the correct position, especially the Vietnamese open-source model. Therefore, we further improve our method by fine-tuning the Marian-NMT models on XQuAD samples in which we include these special delimiters.

Natural language inference datasets are easy and cheap to translate, while passage ranking datasets are more expensive to translate but do not present technical difficulties. We do not recommend automatic translation of question answering datasets due to challenges posed by the task format. However, advances in the methods for translating them may result in better performance for cross-lingual question answering models.

We show that our cross-lingual models can be competitive or even outperform

models fine-tuned on datasets originally created in the target language, especially for the tasks of question answering and natural language inference in Portuguese. This shows that, in the absence of resources in a target language, a cross-lingual method can be a good solution.

We also demonstrate that fine-tuning on both original and translated datasets offers extra gains for natural language inference and passage ranking tasks. With this, we improve the state of the art by 2 points on ASSIN2. On FaQuAD, our zero-shot method is over 20 F1 points above the performance of the model fine-tuned on the target data.

Based on our results, we conclude that the best cross-lingual method is highly dependent on the task, deployment requirements, and available financial resources. The main takeaway from our study is that it is possible to perform well on a task in a target language without necessarily having labeled data in that language.

5.2.2 Cross-domain

In the cross-domain study, we explore the zero-shot transfer ability to the legal domain. We show that for the legal case entailment task, pretrained language models without any fine-tuning on the target task and target domain can perform better than models fine-tuned on the task itself. We also confirm a counter-intuitive result: that models with no adaptation to the target task can have better generalization abilities than models that have been carefully fine-tuned to the task at hand. Furthermore, we show that there is room for improvement in zero-shot performance if larger models are used.

It should also be noted that our research may have implications for future experiments beyond the scope of the legal case entailment task and it is possible that other legal tasks with limited labeled data, such as legal question answering, could also benefit from our zero-shot method.

5.3 Future directions

In this section, we provide a glimpse into the future of the field, especially with regard to the transfer learning categories discussed in this dissertation.

The field of NLP has gone through great technological advances, unfortunately the vast majority of them have been done in just a few high-resource languages (JOSHI *et al.*, 2020). As we discussed earlier, this disparity between research and the real world has important implications. As we have over 6,000 languages spoken worldwide and the top 10 dominant languages represent over 80% of the total internet content, it is critical to expand non-English resources and bring NLP technologies into the rest of languages, since minority language speakers, especially non-Western languages, are being limited in

their access to technology. Therefore, extending the cross-lingual benchmarks to more languages and tasks is an important way forward in future research.

Even for languages that are not dominant but well represented on the internet, they suffer from a scarcity of labeled data to train NLP models. To alleviate this problem, multilingual models and cross-lingual learning methods play an important role in bridging this language divide. By combining the two methods, we can leverage the resources available in dominant languages to develop high-performance NLP systems for low-resource languages. One research direction is to develop larger models to take advantage of their ability to potentially store more languages in their parameters. An alternative research direction to address the lack of labeled data is to improve unsupervised cross-lingual methods, since, in comparison, unlabeled data is much more available.

Current NLP models are still unstable in many scenarios that involve different domains between training and inference data. The development of models to overcome this lack of generalization, especially in data scarcity scenarios, is an important challenge to be faced in future research. Thus, new cross-domain approaches along with more robust models, capable of building common representation spaces across different domains, is a path to be pursued.

Given that the low availability of labeled datasets is a serious weakness in NLP and choosing good source datasets for zero-shot scenarios is an open problem, in the long run it is desirable for our models to be able to take advantage of previously learned information from related domains, languages and tasks to learn complex relationships about the world and generalize well to a wide variety of new scenarios, similar to what humans do.

Bibliography

ARTETXE, M.; LABAKA, G.; AGIRRE, E. Translation artifacts in cross-lingual transfer learning. *arXiv preprint arXiv:2004.04721*, 2020. Cited 2 times on pages 48 and 50.

ARTETXE, M.; RUDER, S.; YOGATAMA, D. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019. Cited 3 times on pages 40, 43, and 49.

BA, J. L.; KIROS, J. R.; HINTON, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. Cited 2 times on pages 30 and 31.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2016. Cited 2 times on pages 26 and 28.

BAJAJ, P.; CAMPOS, D.; CRASWELL, N.; DENG, L.; GAO, J.; LIU, X.; MAJUMDER, R.; MCNAMARA, A.; MITRA, B.; NGUYEN, T.; ROSENBER, M.; SONG, X.; STOICA, A.; TIWARY, S.; WANG, T. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2018. Cited 3 times on pages 44, 47, and 60.

BAMBROO, P.; AWASTHI, A. Legaldb: Long distilbert for legal document classification. In: IEEE. *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. [S.l.], 2021. p. 1–4. Cited on page 58.

BAO, H.; DONG, L.; WEI, F.; WANG, W.; YANG, N.; LIU, X.; WANG, Y.; PIAO, S.; GAO, J.; ZHOU, M.; HON, H.-W. *UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training*. 2020. ArXiv. Disponível em: <https://www.microsoft.com/en-us/research/publication/unilmv2-pseudo-masked-language-models-for-unified-language-model-pre-training/>. Cited on page 58.

BENDRE, N.; MARíN, H. T.; NAJAFIRAD, P. Learning from few samples: A survey. *arXiv preprint arXiv:2007.15484*, 2020. Cited on page 36.

BENGIO, Y.; DUCHARME, R.; VINCENT, P. A neural probabilistic language model. *Advances in Neural Information Processing Systems 13*, 2001. Cited on page 24.

BONIFACIO, L. H.; CAMPIOTTI, I.; LOTUFO, R. de A.; NOGUEIRA, R. mmarco: A multilingual version of MS MARCO passage ranking dataset. *CoRR*, abs/2108.13897, 2021. Disponível em: <https://arxiv.org/abs/2108.13897>. Cited 3 times on pages 47, 50, and 56.

BROWN, T.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J. D.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.;

ZIEGLER, D.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. Language models are few-shot learners. In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M. F.; LIN, H. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020. v. 33, p. 1877–1901. Disponível em: <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf>. Cited 3 times on pages 36, 58, and 66.

CABEZUDO, M. A. S.; INáCIO, M.; RODRIGUES, A. C.; CASANOVA, E.; SOUSA, R. F. de. Natural language inference for portuguese using bert and multilingual information. *Computational Processing of the Portuguese Language. Lecture Notes in Computer Science*, 2020. Cited on page 41.

CARMO, D.; PIAU, M.; CAMPIOTTI, I.; NOGUEIRA, R.; LOTUFO, R. *PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data*. 2020. Cited on page 56.

CHALKIDIS, I.; KAMPAS, D. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law volume 27, pages171–198(2019)*, 2019. Cited on page 59.

CONNEAU, A.; KHANDELWAL, K.; GOYAL, N.; CHAUDHARY, V.; WENZEK, G.; GUZMáN, F.; GRAVE, E.; OTT, M.; ZETTLEMOYER, L.; STOYANOV, V. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2020. Cited 4 times on pages 18, 37, 40, and 41.

CONNEAU, A.; LAMPLE, G. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 2019. Cited 3 times on pages 36, 40, and 41.

CONNEAU, A.; LAMPLE, G.; RINOTT, R.; WILLIAMS, A.; BOWMAN, S. R.; SCHWENK, H.; STOYANOV, V. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*, 2019. Cited 2 times on pages 44 and 46.

DANDAPAT, S.; BISWAS, P.; CHOUDHURY, M.; BALI, K. Complex linguistic annotation – no easy way out! a case from bangla and hindi pos labeling tasks. *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, 2009. Cited on page 40.

DASH, T.; CHITLANGIA, S.; AHUJA, A.; SRINIVASAN, A. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *arXiv preprint arXiv:2107.10295*, 2021. Cited on page 38.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255. Cited on page 35.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Cited 4 times on pages 35, 36, 40, and 42.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the*

2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). [S.l.: s.n.], 2019. p. 4171–4186. Cited on page 58.

ELNAGGAR, A.; GEBENDORFER, C.; GLASER, I.; MATTHES, F. Multi-task deep learning for legal document translation, summarization and multi-label classification. *AICCC '18: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference December 2018 Pages 9–15*, 2018. Cited on page 59.

ELNAGGAR, A.; WALTL, B.; GLASER, I.; LANDTHALER, J.; SCEPANKOVA, E.; MATTHES, F. Stop illegal comments: A multi-task deep learning approach. *AICCC '18: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference December 2018 Pages 41–47*, 2018. Cited on page 59.

ELWANY, E.; MOORE, D.; OBEROI, G. BERT goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding. In: *Workshop on Document Intelligence at NeurIPS 2019*. [S.l.: s.n.], 2019. Cited on page 58.

FARAHANI, A.; POURSHOJAE, B.; RASHEED, K.; ARABNIAN, H. R. A concise review of transfer learning. *arXiv preprint arXiv:2104.02144*, 2021. Cited on page 34.

FARAHANI, A.; VOGHOEI, S.; RASHEED, K.; ARABNIA, H. R. A brief review of domain adaptation. *arXiv preprint arXiv:2010.03978*, 2020. Cited on page 38.

FILHO, J. A. W.; WILKENS, R.; IDIART, M.; VILLAVICENCIO, A. The brwac corpus: A new open resource for brazilian portuguese. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. Cited on page 42.

FONSECA, E.; SANTOS, L.; CRISCUOLO, M.; ALUISIO, S. Assin: avaliacao de similaridade semantica e inferencia textual. *Computational Processing of the Portuguese Language - 12th International Conference, Tomar, Portugal, 13–15*, 2016. Cited on page 44.

GAO, L.; DAI, Z.; CALLAN, J. Rethink training of bert rerankers in multi-stage retrieval pipeline. *arXiv preprint arXiv:2101.08751*, 2021. Cited on page 58.

GOYAL, N.; DU, J.; OTT, M.; ANANTHARAMAN, G.; CONNEAU, A. Larger-scale transformers for multilingual masked language modeling. *arXiv e-prints*, p. arXiv–2105, 2021. Cited on page 41.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. Cited on page 30.

HE, P.; GAO, J.; CHEN, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021. Cited on page 31.

HE, P.; LIU, X.; GAO, J.; CHEN, W. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2020. Cited 3 times on pages 31, 58, and 60.

HEATON, J. Applications of deep neural networks. *arXiv preprint arXiv:2009.05673*, 2020. Cited on page 24.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation 9(8):1735-80*, 1997. Cited on page 25.

HUANG, X.; MAY, J.; PENG, N. What matters for neural cross-lingual named entity recognition: An empirical analysis. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. [S.l.: s.n.], 2019. p. 6396–6402. Cited 2 times on pages 17 and 40.

ISBISTER, T.; CARLSSON, F.; SAHLGREN, M. Should we stop training more monolingual models, and simply use machine translation instead? In: . [S.l.: s.n.], 2021. Cited on page 41.

JORDAN, M. I. Serial order: a parallel distributed processing approach. *Tech. rep. ICS 8604*, 1986. Cited on page 25.

JOSHI, P.; SANTY, S.; BUDHIRAJA, A.; BALI, K.; CHOUDHURY, M. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*, 2020. Cited on page 70.

JUNCZYS-DOWMUNT, M.; GRUNDKIEWICZ, R.; DWOJAK, T.; HOANG, H.; HEAFIELD, K.; NECKERMANN, T.; SEIDE, F.; GERMANN, U.; AJI, A. F.; BOGOYCHEV, N.; MARTINS, A. F. T.; BIRCH, A. Marian: Fast neural machine translation in C++. In: *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 116–121. Disponível em: <http://www.aclweb.org/anthology/P18-4020>. Cited on page 48.

K, K.; WANG, Z.; MAYHEW, S.; ROTH, D. Cross-lingual ability of multilingual bert: An empirical study. *arXiv preprint arXiv:1912.07840*, 2019. Cited on page 40.

KANO, Y.; KIM, M.; GOEBEL, R.; SATOH, K. Overview of COLIEE 2017. In: *COLIEE 2017 (EPiC Series in Computing, vol. 47)*. [S.l.: s.n.], 2017. p. 1–8. Cited on page 60.

KANO, Y.; KIM, M.-Y.; YOSHIOKA, M.; LU, Y.; RABELO, J.; KIYOTA, N.; GOEBEL, R.; SATOH, K. COLIEE-2018: Evaluation of the competition on legal information extraction and entailment. In: *JSAI International Symposium on Artificial Intelligence*. [S.l.: s.n.], 2018. p. 177–192. Cited on page 60.

KHASHABI, D.; MIN, S.; KHOT, T.; SABHARWAL, A.; TAFJORD, O.; CLARK, P.; HAJISHIRZI, H. Unifiedqa: Crossing format boundaries with a single qa system. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. [S.l.: s.n.], 2020. p. 1896–1907. Cited on page 58.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*, 2014. Cited on page 52.

KUDO, T.; RICHARDSON, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018. Cited on page 27.

KWIATKOWSK; PALOMAKI; REDFIELD. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 2019. Cited on page 45.

LEIVADITI, S.; ROSSI, J.; KANOULAS, E. A benchmark for lease contract review. *arXiv preprint arXiv:2010.10386*, 2020. Cited on page 58.

LEPIKHIN, D.; LEE, H.; XU, Y.; CHEN, D.; FIRAT, O.; HUANG, Y.; KRIKUN, M.; SHAZEER, N.; CHEN, Z. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. Cited on page 40.

LEWIS, M.; LIU, Y.; GOYAL, N.; GHAZVININEJAD, M.; MOHAMED, A.; LEVY, O.; STOYANOV, V.; ZETTLEMOYER, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2020. p. 7871–7880. Cited on page 58.

LIN, J.; MA, X.; LIN, S.-C.; YANG, J.-H.; PRADEEP, R.; NOGUEIRA, R. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*, 2021. Cited on page 42.

LIN, J.; NOGUEIRA, R.; YATES, A. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467*, 2020. Cited on page 60.

LIN, T.; WANG, Y.; LIU, X.; QIU, X. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021. Cited on page 31.

LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTLEMOYER, L.; STOYANOV, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. Cited on page 60.

LU, J.; GONG, P.; YE, J.; ZHANG, C. Learning from very few samples: A survey. *arXiv preprint arXiv:2009.02653*, 2020. Cited 2 times on pages 36 and 58.

LUONG, M.-T.; PHAM, H.; MANNING, C. D. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. Cited on page 28.

MA, X.; GUO, J.; ZHANG, R.; FAN, Y.; JI, X.; CHENG, X. Prop: Pre-training with representative words prediction for ad-hoc retrieval. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. [S.l.: s.n.], 2021. p. 283–291. Cited on page 58.

MIKOLOV, T. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013. Cited 2 times on pages 27 and 59.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Cited 3 times on pages 25, 27, and 59.

MOLLER, T.; RISCH, J.; PIETSCH, M. Germanquad and germandpr:improving non-english question answering and passage retrieval. *arXiv preprint arXiv:2104.12741*, 2021. Cited on page 45.

NGUYEN, H.-T.; VUONG, H.-Y. T.; NGUYEN, P. M.; DANG, B. T.; BUI, Q. M.; VU, S. T.; NGUYEN, C. M.; TRAN, V.; SATOH, K.; NGUYEN, M. L. Jnlp team: Deep learning for legal processing in coliee 2020. *arXiv preprint arXiv:2011.08071*, 2020. Cited 2 times on pages 59 and 65.

NGUYEN, K. V.; NGUYEN, D.-V.; NGUYEN, A. G.-T.; NGUYEN, N. L.-T. A vietnamese dataset for evaluating machine reading comprehension. *arXiv preprint arXiv:2009.14725*, 2020. Cited on page 46.

NOGUEIRA, R.; JIANG, Z.; PRADEEP, R.; LIN, J. Document ranking with a pretrained sequence-to-sequence model. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. [S.l.: s.n.], 2020. p. 708–718. Cited 4 times on pages 52, 56, 60, and 63.

NOZZA, D.; FERSINI, E.; MESSINA, E. Deep learning and ensemble methods for domain adaptation. *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2016. Cited on page 17.

OTTER, D. W.; MEDINA, J. R.; KALITA, J. K. A survey of the usages of deep learning in natural language processing. *arXiv preprint arXiv:1807.10854*, 2018. Cited on page 24.

PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (Volume: 22, Issue: 10, Oct. 2010)*, 2010. Cited on page 35.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Disponível em: <https://www.aclweb.org/anthology/D14-1162>. Cited on page 25.

PETERS, M. E.; RUDER, S.; SMITH, N. A. To tune or not to tune? adapting pretrained representations to diverse tasks. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. [S.l.: s.n.], 2019. p. 7–14. Cited on page 58.

PHANG, J.; CALIXTO, I.; HTUT, P. M.; PRUKSACHATKUN, Y.; LIU, H.; VANIA, C.; KANN, K.; BOWMAN, S. R. English intermediate-task training improves zero-shot cross-lingual transfer too. *arXiv preprint arXiv:2005.13013*, 2020. Cited on page 40.

PIETIKäINEN, M.; SILVEN, O. Challenges of artificial intelligence – from machine learning and computer vision to emotional intelligence. *arXiv preprint arXiv:2201.01466*, 2022. Cited on page 22.

PIRES, R.; SOUZA, F.; ROSA, G.; NOGUEIRA, R.; LOTUFO, R. Sequence-to-sequence models for extracting information from registration and legal documents. *arXiv preprint arXiv:2201.05658*, 2022. Cited on page 21.

PIRES, T.; SCHLINGER, E.; GARRETTE, D. How multilingual is multilingual bert? *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4996– 5001, Florence, Italy. Association for Computational Linguistics.*, 2019. Cited on page 40.

POURPANAH, F.; ABDAR, M.; LUO, Y.; ZHOU, X.; WANG, R.; LIM, C. P.; WANG, X.-Z. A review of generalized zero-shot learning methods. *arXiv preprint arXiv:2011.08641*, 2020. Cited on page 36.

PRADEEP, R.; MA, X.; ZHANG, X.; CUI, H.; XU, R.; NOGUEIRA, R.; LIN, J. H2oloo at trec 2020: When all you got is a hammer... deep learning, health misinformation, and precision medicine. *Corpus*, v. 5, n. d3, p. d2, 2020. Cited 4 times on pages 19, 59, 60, and 65.

RABELO, J.; GOEBEL, R.; KIM, M.-Y.; YOSHIOKA, M.; KANO, Y.; SATOH, K. Summary of the competition on legal information extraction/entailment (coliee) 2021. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment*, 2021. Cited on page 20.

RABELO, J.; GOEBEL, R.; KIM, M.-Y.; KANO, Y.; YOSHIOKA, M.; SATOH, K. Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. *Rev Socionetwork Strat (2022). https://doi.org/10.1007/s12626-022-00105-z*, 2022. Cited on page 61.

RABELO, J.; KIM, M.; GOEBEL, R. Application of text entailment techniques in coliee 2020. *International Workshop on Juris-informatics (JURISIN) associated with JSAI International Symposia on AI (JSAI-isAI)*, 2020. Cited on page 59.

RABELO, J.; KIM, M.-Y.; GOEBEL, R. Combining similarity and transformer methods for case law entailment. In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL '19)*. [S.l.: s.n.], 2019. p. 290–296. Cited on page 58.

RABELO, J.; KIM, M.-Y.; GOEBEL, R.; YOSHIOKA, M.; KANO, Y.; SATOH, K. A summary of the COLIEE 2019 competition. In: *JSAI International Symposium on Artificial Intelligence*. [S.l.: s.n.], 2019. p. 34–49. Cited on page 60.

RABELO, J.; KIM, M.-Y.; GOEBEL, R.; YOSHIOKA, M.; KANO, Y.; SATOH, K. Coliee 2020: Methods for legal document retrieval and entailment. 2020. Cited 2 times on pages 59 and 60.

RADFORD, A.; KIM, J. W.; HALLACY, C.; RAMESH, A.; GOH, G.; AGARWAL, S.; SASTRY, G.; ASKELL, A.; MISHKIN, P.; CLARK, J. *et al.* Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. Cited on page 59.

RAFFEL, C.; SHAZEER, N.; ROBERTS, A.; LEE, K.; NARANG, S.; MATENA, M.; ZHOU, Y.; LI, W.; LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. Cited on page 43.

RAFFEL, C.; SHAZEER, N.; ROBERTS, A.; LEE, K.; NARANG, S.; MATENA, M.; ZHOU, Y.; LI, W.; LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, v. 21, n. 140, p. 1–67, 2020. Disponível em: <http://jmlr.org/papers/v21/20-074.html>. Cited 4 times on pages 19, 58, 59, and 60.

RAJPURKAR, P.; ZHANG, J.; LOPYREV, P. L. K. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016. Cited 2 times on pages 43 and 45.

RAMPONI, A.; PLANK, B. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*, 2020. Cited on page 17.

REAL, L.; FONSECA, E.; OLIVEIRA, H. G. The assin 2 shared task: A quick overview. *Computational Processing of the Portuguese Language. Lecture Notes in Computer Science*, 2020. Cited on page 47.

ROBERTS, K.; DEMNER-FUSHMAN, D.; VOORHEES, E.; HERSH, W.; BEDRICK, S.; LAZAR, A. J.; PANT, S. Overview of the trec 2019 precision medicine track. *The ... text REtrieval conference : TREC. Text REtrieval Conference*, v. 26, 2019. Cited 2 times on pages 19 and 59.

ROBERTSON, S. E.; WALKER, S.; JONES, S.; HANCOCK-BEAULIEU, M. M.; GATFORD, M. *et al.* Okapi at trec-3. *Nist Special Publication Sp*, NATIONAL INSTIUTE OF STANDARDS & TECHNOLOGY, v. 109, p. 109, 1995. Cited on page 59.

RODRIGUES, R.; COUTO, P.; RODRIGUES, I. Ipr: The semantic textual similarity and recognizing textual entailment systems. *Conference: Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil,*, 2019. Cited on page 56.

RODRIGUES, R.; SILVA, J. da; CASTRO, P.; FELIX, N.; SOARES, A. Multilingual transformer ensembles for portuguese natural language tasks. *Conference: ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in PortugueseAt: http://ceur-ws.org/Vol-2583/Volume: Vol-2583*, 2020. Cited 2 times on pages 41 and 56.

ROSA, G. M.; BONIFACIO, L. H.; SOUZA, L. R. de; LOTUFO, R.; NOGUEIRA, R. A cost-benefit analysis of cross-lingual transfer methods. *arXiv preprint arXiv:2105.06813*, 2021. Cited on page 20.

ROSA, G. M.; RODRIGUES, R. C.; ; NOGUEIRA, R.; LOTUFO, R. Yes, bm25 is a strong baseline for legal case retrieval. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment*, 2021. Cited on page 21.

ROSA, G. M.; RODRIGUES, R. C.; LOTUFO, R.; NOGUEIRA, R. To tune or not to tune? zero-shot models for legal case entailment. *ICAIL'21, Eighteenth International Conference on Artificial Intelligence and Law, June 21–25, 2021, São Paulo, Brazil*, 2021. Cited on page 20.

RUDER, S. *Neural Transfer Learning for Natural Language Processing*. Tese (Doutorado) — National University of Ireland, Galway, 2019. Cited on page 36.

RUDER, S.; VULIć, I.; SøGAARD, A. A survey of cross-lingual word embedding models. *arXiv preprint arXiv:1706.04902*, 2017. Cited on page 37.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature. 323 (6088): 533–536.*, 1986. Cited on page 33.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach (AIMA)*. [S.l.]: Pearson, 2020. Cited on page 22.

SABOU, M.; BONTCHEVA, K.; SCHARL, A. Crowdsourcing research opportunities: lessons from natural language processing. *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies, pages 1–8*, 2012. Cited on page 40.

SAGEL, A.; SAHU, A.; MATTHES, S.; PFEIFER, H.; QIU, T.; RUEß, H.; SHEN, H.; WöRMANN, J. Knowledge as invariance – history and perspectives of knowledge-augmented machine learning. *arXiv preprint arXiv:2012.11406*, 2020. Cited on page 34.

SAYAMA, H. F.; ARAUJO, A. V.; FERNANDES, E. R. Faquad: Reading comprehension dataset in the domain of brazilian higher education. *arXiv preprint arXiv:1906.05743*, 2019. Cited on page 45.

SCHICK, T.; SCHÜTZE, H. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020. Cited 2 times on pages 36 and 58.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *arXiv preprint arXiv:1404.7828*, 2014. Cited on page 23.

SCHUSTER, M.; NAKAJIMA, K. Japanese and korean voice search. In: IEEE. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2012. p. 5149–5152. Cited on page 27.

SHAGHAGHIAN, S.; FENG, L. Y.; JAFARPOUR, B.; POGREBNYAKOV, N. Customizing contextualized language models for legal document reviews. In: IEEE. *2020 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2020. p. 2139–2148. Cited on page 58.

SHAZEER, N.; STERN, M. Adafactor: Adaptive learning rates with sublinear memory cost. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2018. p. 4596–4604. Cited on page 52.

SHI, S.; BAI, H.; LIN, J. Cross-lingual training of neural models for document ranking. *Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2768-2773, November 2020*, 2020. Cited on page 41.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: Pretrained bert models for brazilian portuguese. *9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I*, 2020. Cited 2 times on pages 42 and 56.

SOUZA, L. R. de; NOGUEIRA, R.; LOTUFO, R. On the ability of monolingual models to learn language-agnostic representations. *arXiv preprint arXiv:2109.01942*, 2021. Cited on page 54.

STEINBERGER, R.; POULIQUEN, B.; WIDIGER, A.; IGNAT, C.; ERJAVEC, T.; TUFIş, D.; VARGA4, D. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. 2006. Cited on page 37.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014. Cited on page 25.

TAM, D.; MENON, R. R.; BANSAL, M.; SRIVASTAVA, S.; RAFFEL, C. Improving and simplifying pattern exploiting training. *arXiv preprint arXiv:2103.11955*, 2021. Cited 2 times on pages 36 and 58.

TAY, Y.; DEHGHANI, M.; BAHRI, D.; METZLER, D. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020. Cited 2 times on pages and 26.

THAKUR, N.; REIMERS, N.; RüCKLé, A.; SRIVASTAVA, A.; GUREVYCH, I. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 4 2021. Disponível em: <https://arxiv.org/abs/2104.08663>. Cited 2 times on pages 19 and 59.

TIEDEMANN, J.; THOTTINGAL, S. OPUS-MT — Building open translation services for the World. In: *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*. Lisbon, Portugal: [s.n.], 2020. Cited on page 48.

TOOSI, A.; BOTTINO, A.; SABOURY, B.; SIEGEL, E.; RAHMIM, A. A brief history of ai: how to prevent another winter (a critical review). *arXiv preprint arXiv:2109.01517*, 2021. Cited on page 22.

TORFI, A.; SHIRVANI, R. A.; KENESHLOO, Y.; TAVAF, N.; FOX, E. A. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*, 2020. Cited on page 24.

TRISCHLER, A.; WANG, T.; YUAN, X.; HARRIS, J.; SORDONI, A.; BACHMAN, P.; SULEMAN, K. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1901.08634*, 2017. Cited on page 43.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2017. p. 5998–6008. Cited 6 times on pages 26, 31, 33, 42, 43, and 58.

VOORHEES, E. M. Overview of the trec 2004 robust track. *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, November 16-19, 2004*, 2004. Cited 2 times on pages 44 and 60.

WANG, H.; RAJ, B. On the origin of deep learning. *arXiv preprint arXiv:1702.07800*, 2017. Cited on page 23.

WEI, J.; BOSMA, M.; ZHAO, V. Y.; GUU, K.; YU, A. W.; LESTER, B.; DU, N.; DAI, A. M.; LE, Q. V. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. Cited on page 66.

WILLIAMS, A.; NANGIA, N.; BOWMAN, S. A broad-coverage challenge corpus for sentence understanding through inference. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. p. 1112–1122. Disponível em: <http://aclweb.org/anthology/N18-1101>. Cited 2 times on pages 44 and 46.

WOLF, T.; DEBUT, L.; SANH, V.; CHAUMOND, J.; DELANGUE, C.; MOI, A.; CISTAC, P.; RAULT, T.; LOUF, R.; FUNTOWICZ, M.; DAVISON, J.; SHLEIFER, S.; PLATEN, P. von; MA, C.; JERNITE, Y.; PLU, J.; XU, C.; SCAO, T. L.; GUGGER, S.; DRAME, M.; LHOEST, Q.; RUSH, A. M. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. Cited 2 times on pages 48 and 52.

WU, M.; MOOSAVI, N. S.; RüCKLé, A.; GUREVYCH, I. Improving qa generalization by concurrent modeling of multiple biases. *arXiv preprint arXiv:2010.03338*, 2020. Cited on page 17.

WU, S.; DREDZE, M. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*, 2019. Cited 2 times on pages 18 and 40.

WU, Y.; SCHUSTER, M.; CHEN, Z.; LE, Q. V.; NOROUZI, M.; MACHEREY, W.; KRIKUN, M.; CAO, Y.; GAO, Q.; MACHEREY, K. *et al.* Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. Cited on page 40.

XIAO, C.; ZHONG, H.; GUO, Z.; TU, C.; LIU, Z.; SUN, M.; FENG, Y.; HAN, X.; HU, Z.; WANG, H.; XU, J. CAIL2018: A large-scale legal dataset for judgment prediction. *arXiv:1807.02478*, 2018. Cited on page 58.

XUE, L.; CONSTANT, N.; ROBERTS, A.; KALE, M.; AL-RFOU, R.; SIDDHANT, A.; BARUA, A.; RAFFEL, C. *mT5: A massively multilingual pre-trained text-to-text transformer*. 2021. Cited 5 times on pages 18, 40, 41, 43, and 52.

YANG, P.; FANG, H.; LIN, J. Anserini: Enabling the use of lucene for information retrieval research. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2017. (SIGIR '17), p. 1253–1256. ISBN 9781450350228. Disponível em: <https://doi.org/10.1145/3077136.3080721>. Cited on page 53.

YEUNG, C. M. *Effects of inserting domain vocabulary and fine-tuning BERT for German legal language*. Dissertação (Mestrado) — University of Twente, 2019. Cited on page 58.

YIN, W.; HAY, J.; ROTH, D. *Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach*. 2019. Cited on page 58.

YIN, W.; RAJANI, N. F.; RADEV, D.; SOCHER, R.; XIONG, C. *Universal Natural Language Processing with Limited Annotations: Try Few-shot Textual Entailment as a Start*. 2020. Cited on page 58.

YIN, W.; SCHÜTZE, H.; XIANG, B.; ZHOU, B. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, MIT Press, v. 4, p. 259–272, 2016. Cited on page 58.

ZHANG, E.; GUPTA, N.; NOGUEIRA, R.; CHO, K.; LIN, J. Rapidly deploying a neural search engine for the covid-19 open research dataset. In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. [S.l.: s.n.], 2020. Cited 2 times on pages 19 and 59.

ZHONG, H.; XIAO, C.; TU, C.; ZHANG, T.; LIU, Z.; SUN, M. How does NLP benefit legal system: A summary of legal artificial intelligence. *arXiv:2004.12158*, 2020. Cited 2 times on pages 58 and 59.

ZHONG, H.; XIAO, C.; TU, C.; ZHANG, T.; LIU, Z.; SUN1, M. Jec-qa: A legal-domain question answering dataset. *Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 9701-9708.*, 2020. Cited on page 59.

ZHUANG, F.; QI, Z.; DUAN, K.; XI, D.; ZHU, Y.; ZHU, H.; XIONG, H.; HE, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE, 1–34. doi:10.1109/jproc.2020.3004555* , 2020. Cited on page 35.