TÉCNICO LISBOA

Simultaneous Tagging of Named Entities and Parts-of-Speech for Portuguese and Spanish Texts

Luís Miguel Paiva Sampaio Santos

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Bruno Emanuel da Graça Martins
Dr. Pedro Paulo Balage Filho

Examination Committee

Chairperson: Prof. Paolo Romano
Supervisor: Prof. Bruno Emanuel da Graça Martins
Member of the Committee: Dr. Erick Rocha Fonseca

June 2019

Acknowledgments

I would first like to thank both my advisors, Prof. Bruno Martins and Dr. Pedro Balage, for all the guidance, patience and support throughout all the phases of my thesis. They were always accessible whenever I had a question about my research or writing, and only with their constant feedback could I ever hope to complete this work.

I would also like to express my gratitude to Prof. Nuno Mamede for his insights on handling contractions for the Portuguese language, and Rita Marquilhas from CLUL for her help on parsing the XML contents of the Post Scriptum corpus.

To my family, especially my parents, for their unconditional love and support during all these years that led me to this point. To my friends who have been with me from the start through this hard-earned but nevertheless rewarding academic journey. This accomplishment would not be possible without all of your support. Thank you.

Abstract

Named entity recognition and parts-of-speech tagging are fundamental tasks in the field of natural language processing, currently with many practical applications. The current state-of-the-art approaches are based on the supervised training of deep neural networks, achieving near-human level accuracy. However, on less-resource scenarios arising from processing historical texts or languages other than English, the fact that few training corpora exist limits the use of modern machine learning approaches. To address this limitation, we collected and standardized a wide variety of datasets containing text in Portuguese and Spanish, annotated according to parts-of-speech and/or named entities. We then evaluated a modern neural architecture for sequence labeling, considering transfer learning approaches based on multi-task learning (i.e., simultaneously addressing parts-of-speech tagging and named entity recognition) and cross-lingual learning (i.e., aligning word embeddings of the Portuguese and Spanish languages in a single vector space), in order to exploit all the available data and the underlying similarities on these tasks/languages, specifically to improve generalization on the smaller historical datasets. Our cross-lingual model, i.e. a joint approach for annotating texts with parts-of-speech and named entities in Portuguese and Spanish, achieves 91.97% of POS accuracy and 84.60% of entity-level F1 score for Portuguese, and 93.91% of POS accuracy and 64.34% of entity-level F1 score for Spanish, when averaging over all datasets for these languages. We also release a collection of 13 standardized datasets to the research community to further stimulate research in these understudied languages and domains.

Keywords

Natural Language Processing; Deep Learning Models; Named Entity Recognition; Multi-Task Learning; Cross-Language Learning

Resumo

O reconhecimento de entidades mencionadas e a etiquetação morfo-sintática são tarefas fundamentais na área de processamento de língua natural, atualmente com diversas aplicações práticas. O estado de arte nestas tarefas consiste no treino supervisionado de redes neuronais profundas, alcançando resultados próximos a peritos humanos nestas tarefas. No entanto, em cenários com menos recursos, como o processamento de textos históricos ou em línguas diferentes do Inglês, o facto que existem poucos corpora de treino limita a aplicação de técnicas modernas para aprendizagem automática. Para combater esta limitação, compilámos e normalizámos uma lista exaustiva de corpora contendo texto em Português e em Espanhol, anotado com categorias morfo-sintáticas e/ou entidades mencionadas. Posteriormente, avaliámos uma arquitetura neuronal moderna para etiquetação de sequências, considerando técnicas de aprendizagem por transferência baseadas em aprendizagem multitarefa, aprendendo simultaneamente nas tarefas de etiquetagem morfo-sintática e reconhecimento de entidades mencionadas, e aprendizagem multilingue, alinhando os embeddings de Português e Espanhol num espaço vetorial comum. Esta abordagem permite tirar partido de todos os dados disponíveis de forma a explorar semelhanças subjacentes nestas tarefas/línguas, com o intuito de melhorar a performance em textos históricos. O nosso modelo multilingue, i.e. uma abordagem unificada para anotar textos com categorias morfo-sintáticas e entidades mencionadas em Português e Espanhol, alcança 91.97% de exactidão e 84.60% de F1 nas duas tarefas em Português, e 93.91% de exactidão e 64.34% de F1 em Espanhol, ao avaliar em média para todos os dados destas línguas.

Palavras Chave

Processamento de Língua Natural; Modelos de Aprendizagem Profunda; Reconhecimento de Entidades Mencionadas; Aprendizagem Multitarefa; Aprendizagem Multilingue

Contents

List of Figures

List of Tables

1

Introduction

Contents

Leveraging transferable representations is a recurring topic in Natural Language Processing (NLP), popularized in the neural network literature with the advent of word embeddings (Mikolov et al., 2013) and, more recently, with contextual word embeddings (Peters et al., 2018). Other instances of transfer learning include multi-task learning, which exploits underlying similarities among tasks in an attempt to solve multiple related tasks using a single unified neural architecture, or cross-lingual learning, which allows for a neural model to reason in multiple languages, commonly by aligning the word embeddings of each language in a single vector space. This M.Sc. thesis explored recently proposed ideas on multi-task and cross-lingual learning, in an attempt to learn neural models simultaneously on the tasks of parts-of-speech tagging and named entity recognition, and for the Portuguese and Spanish languages.

1.1 Motivation

Together with the recent rise of deep learning methods for NLP (Goldberg, 2017), much emphasis has been put on the development of neural models for Parts-of-Speech (POS) tagging or Named Entity Recognition (NER), specifically for high-resource languages such as English. Deep learning models leverage large training corpora consisting of modern newswire text, reaching near-human performance for these languages/domains. However, for other less-resource languages (e.g., Portuguese or Spanish) and/or domains (e.g., historical text), the development of neural methods did not attract similar attention from the machine learning and NLP communities, due to the lack of the necessary volume of data to train these models. Consequently, there have been only few improvements for these languages/domains compared to English, leading to a significantly lower performance in terms of state-of-the-art models.

Transfer learning is a promising research topic to tackle low-resource scenarios such as languages or domains with scarce training corpora. Although there is renewed interest in transfer learning approaches, which due to the advent of high-capacity neural models for NLP could in principle enable joint training approaches combining different resources, in practice most work has focused only on a select number of languages or domains. Up to this moment, few studies on sequence labeling have focused on the Portuguese and Spanish languages, which as closely related languages are thus fertile ground for testing cross-lingual approaches, and on the sequence labeling tasks of parts-of-speech tagging and named entity recognition, which are fundamental NLP tasks with many practical applications. For the aforementioned tasks, some authors reported no improvements from multi-task learning (Collobert et al., 2011; Søgaard and Goldberg, 2016), and we hope to validate if these claims also hold for the Portuguese and Spanish languages using a modern neural architecture. Differently from previous work, we envision the evaluation of transfer learning approaches on a considerably larger number of datasets.

1.2 Thesis Proposal

Given the importance of the subject, and a certain lack of sequence labeling resources in Portuguese and Spanish featuring consistent annotation schemes, this thesis proposes to experimentally validate the usage of transfer learning approaches to jointly learn neural models on the sequence labeling tasks of Parts-of-Speech (POS) tagging and Named Entity Recognition (NER), for texts written in Portuguese and Spanish. In this way, we strive to investigate whether there are advantages to joint training, knowing that smaller datasets on certain domains/languages could perhaps benefit from joint training with larger datasets on other more abundant domains/languages. Using transfer learning, the main objective of this thesis is to learn effective models capable of annotating texts in both languages and in various domains.

Addressing the lack of readily accessible corpora on certain languages/domains, this work attempted to bring together all scattered (openly available) resources of annotated datasets for Spanish and Portuguese, modern and historical, considering both NER and/or POS annotations. I specifically normalized the annotations and merged these different sources, in order to increase the amount of training data. Although there are relatively few resources with NER annotations for both these languages, several datasets with POS annotations have indeed been made available, including corpora of historical contents. Noticing that POS information can naturally inform the identification of named entities (e.g., named entities typically correspond to proper nouns), neural models were trained to simultaneously address the POS tagging and NER tasks, combining all the available information.

We used pre-trained FastText word embeddings (Bojanowski et al., 2017) representing word tokens and character n-grams, this way directly representing all the word tokens that are likely to be present in the different types of text that we envision processing. The word embeddings in both languages were also mapped to a common representation space (Zhou et al., 2019), allowing us to train models capable of processing text in both Spanish and Portuguese. Given the lexical, morphological, and syntactic similarities between these two languages, we argue that training neural models with a combination of different annotated datasets, covering both languages, can bring further advantages in terms of prediction accuracy.

In terms of the actual models that were used in our work, we experimented with a modern neural network architecture (Lample et al., 2016) combining pre-trained word embeddings, Bi-directional Recurrent Neural Networks (BiRNNs), and a final layer inspired on conditional random fields to ensure the coherence of the sequences of annotations that are produced (e.g., modeling the fact that different entity mentions are frequently separated by other tokens that do not belong to named entities). Similar models are nowadays extensively used in the NLP literature (Goldberg, 2017), and we also considered several other advancements from the current state-of-the-art (e.g., penalized hyperbolic tangent activation functions associated to RNN nodes, as described by Eger et al. (2018)).

1.3 Contributions

The main contributions of this work are as follows:

- The collection of a comprehensive list of 13 datasets, encompassing 8 datasets in Portuguese, 3 datasets in Spanish, and 2 datasets both in Portuguese and Spanish. A total of 5 historical datasets were considered, while the remaining 8 consist of modern, mostly newswire, texts.
- Consistent normalization of the collected datasets in terms of data formats, tokenization strategies and annotation guidelines, thus ensuring that different datasets, created in separate research efforts, can be combined into a single large resource to increase the amount of training data. We release the models, their implementation and all dataset standardization scripts[1], and a package with all the standardized datasets is now also available on request for research purposes.
- Evaluation of a modern sequence labeling architecture on all the gathered corpora, supporting different types of transfer learning. We first conduct tests on each dataset/task in isolation, then considering cross-domain learning by combining datasets from different domains and time periods, considering multi-task learning by combining datasets with POS and NER annotations, and considering cross-lingual learning by combining datasets in Portuguese and Spanish. The most general cross-lingual model achieves 91.97% of overall POS accuracy and 84.60% of entity-level F1 score for Portuguese, and 93.91% of overall POS accuracy and 64.34% of entity-level F1 score for Spanish, when averaging over all datasets for these languages.

1.4 Organization of the Document

The rest of this document is organized as follows. Chapter 2 presents fundamental concepts and related work regarding supervised learning for sequence labeling. Chapter 3 details the architecture of the proposed model, and details the datasets that were collected and normalized. Chapter 4 describes the experimental setup, evaluation metrics, and the obtained results of the experiments. Finally, Chapter 5 concludes this dissertation with the main findings of this work, followed by a discussion of possible directions for future work.

[1] http://github.com/luispsantos/seq-labeling-datasets

2

Concepts and Related Work

Contents

This chapter presents the necessary background on neural networks and sequence labeling, which are required in order to understand the research contained in subsequent chapters. Section 2.1 introduces the reader to natural language processing, word representations and neural networks. Section 2.2 presents past work in neural methods addressing the tasks of POS tagging or NER, highlighting studies that address POS tagging or NER over Portuguese or Spanish texts. We additionally review the literature on multi-task and cross-language learning methods, relevant for these languages of interest.

2.1 Fundamental Concepts

We first introduce natural language processing and the tasks of parts-of-speech tagging and named entity recognition. We cover word representations, followed by neural networks, feedforward networks and recurrent networks. We also provide some clarifications on loss functions and backpropagation.

2.1.1 Natural Language Processing

Natural Language Processing (NLP) deals with computerized methods for the handling of natural language text (i.e., text written by humans) available as news articles, legal documents, journal publications, etc. As a field, NLP sits at the intersection of Artificial Intelligence, Computer Science, and Linguistics. In the late 1980s a new paradigm for processing natural language text emerged based on statistical methods. Previous approaches had focused on writing hand-coded rules for processing language, which had the limitation of being heavily domain-dependent and task-specific. Statistical approaches instead automatically learn these rules from a corpora of annotated documents. A key enabler of change was the increased availability of treebanks (i.e., datasets featuring syntactic or semantic information). The reliance on writing hand-coded rules for language gradually shifted to the use of statistical methods, enabling a new generation of applications to be built more tolerant to natural language variation.

Two common sequence labeling tasks are parts-of-speech tagging and named entity recognition. Both of these tasks correspond to the assignment of a sequence of labels to a sequence of words used as input (i.e., each word is assigned a label). Parts-of-Speech (POS) tagging assigns a label category to a given word which identifies the syntactic role of said word in the sentence (e.g., noun, adjective, adverb or verb). A POS tagset corresponds to a collection of tags used for the implementation of a POS tagger. POS tagsets can be classified as coarse-grained or as fine-grained, depending on whether the number of possible labels is small or large, respectively. Fine-grained tagsets consider additional word information based on morphosyntactic properties such as lexical or inflectional features.

Named Entity Recognition (NER) corresponds to the identification of named entities (e.g., names of persons, organizations or locations) from a word sequence. Since a single entity may span multiple tokens, entities are thus mapped to the token-level via an *encoding scheme*. One such popular scheme

is the BIO encoding (i.e., an acronym of begin, inside, outside). For a given entity *X*, the initial word of said entity is marked with the label *B-X*, meaning that the word is the beginning of an entity. For the words following the *B-X* label, these words are marked with the label *I-X* as long as they are inside entity *X*. Words without any kind of entity mention are marked with the label *O* (i.e., such words are outside of any named entity that belongs to a prespecified list of entity types).

2.1.2 Word representations

The choice of word representation is an important topic in NLP. Computers do not process language easily with words in raw form, and instead, a possible approach is to represent words as discrete symbols. A *one-hot* representation maps words to vectors of size V, where V is the number of distinct words in a document or set of documents (i.e., word dictionary size), with the resulting vector containing a 1 at the word index and 0 in all remaining positions. As the vast majority of positions contain 0, a one-hot representation leads to *sparse* vectors.

In recent years, the usage of *word embeddings* has become standard practice in NLP. In contrast to one-hot vectors, word embeddings employ a *dense* representation in a continuous vector space, leading to a vector size typically much smaller than V. Importantly, syntactic and semantic similarity are encoded directly on the dimensions of the word vectors. In contrast, sparse one-hot vectors share no sense of similarity among words, as these are represented orthogonally to each other. When using word embeddings, words tend to have similar vectors if they occur in similar contexts, as measured in terms of Euclidean or cosine distance, allowing for statistical strength to be shared among different words.

The estimation of word embeddings relies on the *distributional hypothesis*, which simply states that *words that occur in similar contexts tend to have similar meanings*. Methods for estimating word embeddings can be generally divided into two groups: count-based or prediction-based. As an example of a count-based approach, we could calculate a word co-occurrence matrix based on a fixed context window and then apply a technique of rank reduction. More concretely, for each word in the document, word counts are accumulated by searching over the neighborhood of a current word and counting each one of the context words once. A method for dimensionality reduction, most commonly Singular Value Decomposition (SVD), would then be applied to the co-occurrences matrix, thus yielding a low-rank word representation which can be used to assess word similarity.

A popular prediction-based approach for training word embeddings over large quantities of unlabeled textual data is Word2Vec (Mikolov et al., 2013). Before Word2Vec, other prediction-based approaches could already obtain a continuous representation of words which encoded word similarity (Bengio et al., 2003; Collobert et al., 2011). Nonetheless, the work of Mikolov et al. (2013) went a long way in making the training process more efficient, enabling word embeddings to be trained on large quantities of text. Rather than a single model, Word2Vec should be seen as a family of models for training word vectors.

Two main algorithms, namely Skip-gram and CBOW, formulate a supervised training objective which consists of predicting words from other words. These algorithms differ in terms of which words are used for prediction (predictors) or being predicted (predictands). Additionally, two efficient training methods, namely hierarchical softmax or negative sampling, result in overall faster training times.

The Skip-gram model considers a window of size W around a center word, such that W words to the left and W words to the right of the center word become context words. The objective of the skip-gram model is then to predict the context words given the center word. For each word in a corpus of size T words, we move a word window throughout the corpus so that it is centered on each word exactly once. We can write our training objective in the log domain as:

$$\mathsf{L}_{\text{Skip-gram}}(\theta) = -\sum_{t=1}^{T} \sum_{\substack{-W \leq p \leq W \\ p \neq 0}} \log \mathsf{Pr}(w_{t+p}|w_t, \theta) \tag{2.1}$$

The training objective corresponds to minimizing loss $\mathsf{L}(\theta)$ in regard to parameters θ. In this formulation, we do not treat context words differently depending on their position p in a word window. We consider a matrix of center word vectors m and a matrix of context word vectors n, both of dimension $V \times D$ where V is the word vocabulary size and D the dimension of the word embedding. We predict a context word w_j from a center word w_c by calculating a dot product between center word vector m_c and context word vector n_j and normalize via softmax over all possible context vectors to obtain a probability distribution.

$$\mathsf{Pr}(w_j|w_c) = \frac{\exp(n_j\, m_c^\intercal)}{\sum_{v=1}^{V} \exp(n_v\, m_c^\intercal)} \tag{2.2}$$

A close inspection at the denominator term of $\mathsf{Pr}(w_j|w_c)$ shows a summation over all words V in the vocabulary. A sum over all words can quickly become the main bottleneck of the model, especially if we consider that the sum must be computed for each training example and that V can be quite large in practice. Negative sampling is a technique for speeding up the process. We consider a set of k negative examples sampled from an unigram distribution $\mathsf{Pr}(w)$, in this context negative samples means words that do not co-occur with the predictor word (i.e., words which are not predictands). The time complexity is reduced since we only sample k negative words and typically $k \ll V$. In the Skip-gram objective $\mathsf{L}(\theta)$, we replace the term $\log \mathsf{Pr}(w_j|w_c)$ with a term $\log \mathsf{S}(w_j, w_c)$ as:

$$\log \mathsf{S}(w_j, w_c) = \log \sigma(n_j\, m_c^\intercal) + \sum_{s \sim Pr(w)} \log \sigma(-n_s\, m_c^\intercal) \tag{2.3}$$

The sigmoid function σ squashes input values elementwise to lie between 0 and 1. Continuous Bag-of-Words (CBOW) is an alternate model where the words used as predictor and predictand are switched around. Instead of predicting context words from a center word like the Skip-gram model, CBOW predicts

a center word from the context words. We calculate an average context vector $\hat{n}_t$ as:

$$\hat{n}_t = \frac{1}{2W} \sum_{\substack{-W \leq p \leq W \\ p \neq 0}} n_{t+p} \tag{2.4}$$

We can apply the bag-of-words assumption by disregarding order p of context words in relation to the center word. The normalization via softmax becomes over all possible center words. The CBOW loss $\mathsf{L}(\theta)$ and probability $\mathsf{Pr}(w_c|\hat{n})$ become:

$$\begin{aligned} \mathsf{L}_{\mathsf{CBOW}}(\theta) &= -\sum_{t=1}^{T} \log \mathsf{Pr}(w_t|\hat{n}_t, \theta) \\ \mathsf{Pr}(w_c|\hat{n}) &= \frac{\exp(m_c\, \hat{n}^\intercal)}{\sum_{v=1}^{V} \exp(m_v\, \hat{n}^\intercal)} \end{aligned} \tag{2.5}$$

An important limitation of Word2Vec is that the training objectives do not consider subword information (i.e., words are the atomic level of representation). FastText (Bojanowski et al., 2017) is a more recent approach for learning word embeddings that in turn considers character-level information as an extension of the skip-gram and CBOW models objectives. Crucially, words are encoded as a bag of character n-grams, whereby word representations are computed as a sum of the character n-gram vectors. As an example, with $n = 3$, the word *junho* is represented by the trigrams <ju, jun, unh, nho, ho> and also the word sequence <junho>. Due to this encoding trick, FastText can in principle create representations for any word, including rare words or unseen words, simply by summing the corresponding n-gram vectors.

2.1.3 Neural Networks

Neural networks are a computational model loosely based on the human brain. Following the biological interpretation, neural networks are composed by a set of artificial neurons, which act as units of computation, and a set of synapses, which act as connections between pairs of neurons. Each connection is associated with a weight, which represents the strength of synapses between adjacent neurons.

As a mathematical model, the computation over a single output neuron can be described as a weighted sum over the input neurons followed by the application of an activation function (see Figure 2.1). Each input neuron x_i is multiplied by a weight w_i, and the weights correspond to parameters of the model that will be tuned during the learning process. A bias term, which is also a model parameter, is used to shift the scalar output from the summation independently from the inputs, and in that sense it has the same role as an intercept term in linear regression models. The activation function introduces nonlinearity in the network. Common activation functions exhibit a squashing behaviour characterized by a sigmoidal or S-shaped function, and some possible choices are shown in Figure 2.2.

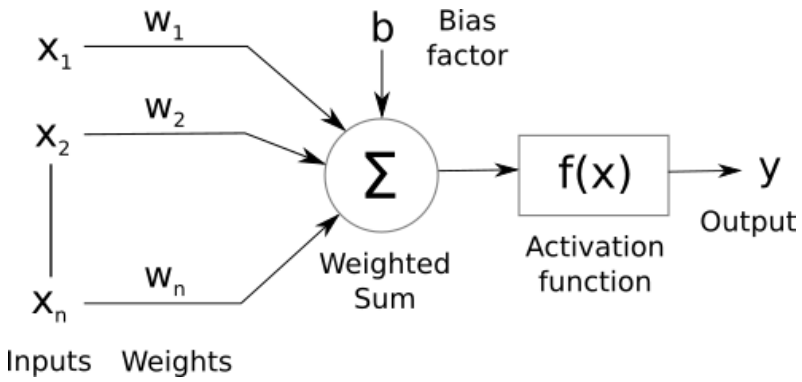

Figure 2.1: Computation for an artificial neuron.

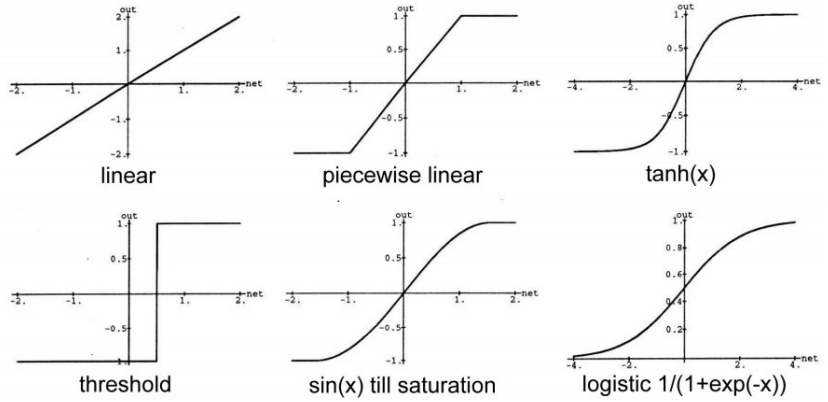

Figure 2.2: Possible choices for activation functions (Graves, 2012).

2.1.4 Feedforward Neural Networks

The computation for the artificial neuron introduced previously deals with a single output neuron over multiple inputs. A more useful model considers multiple input and output neurons organized in a layered architecture. One such Feedforward Neural Network (FNN) is composed by a set of layers (see Figure 2.3), such that neurons from one layer connect only to neurons in the next layer. The first layer is an input layer, which is followed by zero or more hidden layers and a final output layer. Considering an example of a feedforward network with two hidden layers, the following equations describe the computation from the inputs x to the outputs y, where I, H and O denote the sizes of the input, hidden and output layers, respectively, and D denotes the number of datapoints. Activation functions f_1 and f_2 are applied after a linear transformation of the inputs at each layer of the feedforward network.

$$a = f_1(Wx + b_1)$$

$$y = f_2(Ua + b_2)$$

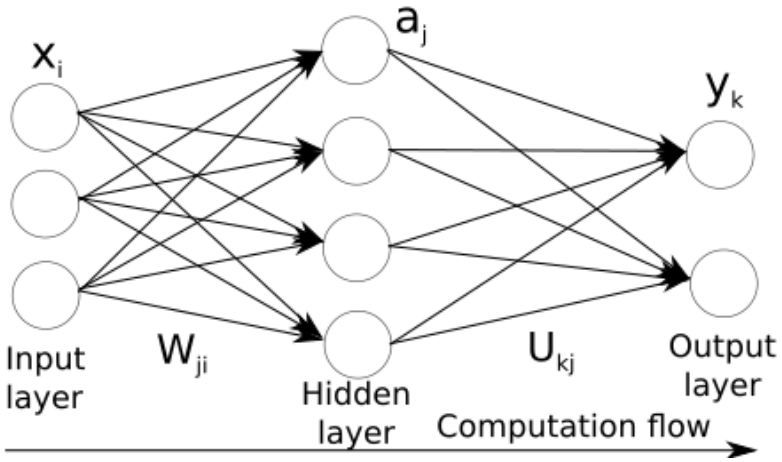

Figure 2.3: Feedforward network with one hidden layer.

$$x \in \mathbb{R}^{I \times D} \qquad W \in \mathbb{R}^{H \times I} \qquad b_1 \in \mathbb{R}^{H}$$
$$y \in \mathbb{R}^{O \times D} \qquad U \in \mathbb{R}^{O \times H} \qquad b_2 \in \mathbb{R}^{O}$$

Feedforward networks with multiple layers are commonly referred to as Multilayer Perceptrons (MLPs). A well known theoretical result states that a feedforward network with just one hidden layer can already approximate any arbitrary function in $\mathbb{R}^N$, this result however says nothing on the optimal hidden layer dimension or how long training would take until convergence.

2.1.5 Loss Functions

A loss function quantifies the amount of error between the network outputs $\hat{y}$ and the real outputs y. It is desirable for $\hat{y}$ and y to be numerically close, as this implies the neural network is doing a good job at predicting y from x. A common loss function for regression tasks is the Mean Squared Error (MSE).

$$\mathsf{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i^2 - \hat{y}_i^2)$$

Learning the best network parameters θ for a specific task is therefore reduced to the problem of minimizing the loss function $\mathsf{L}(\hat{y}, y, \theta)$ defined over the network outputs $\hat{y}$ and the real outputs y. In a classification scenario, it is common to use a 1-of-K encoding to represent the real outputs y. Each class c is represented with a vector y of size K, where K is the number of classes, this vector y consists of a 1 at position $y_{p=c}$ and 0 in all remaining positions $y_{p \neq c}$. As an example taken from the NER task, a single word is classified as being part of a person (PER), organization (ORG) or location (LOC) entity or instead as no entity/other (O).

$$y_{PER} = (1,0,0,0) \qquad y_{LOC} = (0,0,1,0)$$
$$y_{ORG} = (0,1,0,0) \qquad y_{O} = (0,0,0,1)$$

In the previous example, with $K = 4$, we had a small number of classes to predict, but, depending on the task, K may be quite large. For example, in Language Modeling, the task is to predict a word given all the words that appeared before in context, and it is not uncommon for K to be in the order of thousands or even hundreds of thousands. A possible interpretation for 1-of-K encoding is that it creates a discrete probability distribution over the set of possible classes C. The two necessary rules for a valid discrete probability distribution follow:

$$\bigvee_{c \in C} \mathsf{Pr}(c) \geq 0$$

$$\sum_{c \in C} \mathsf{Pr}(c) = 1$$

We can readily confirm that 1-of-K encoding respects the rules to be considered a valid probability distribution. As the real outputs y are represented as a probability distribution, albeit a simple one, by also encoding the network outputs $\hat{y}$ as a probability distribution, the Cross Entropy (CE) loss can be applied. CE is a categorical loss function with its roots on information theory. The softmax function can be used to map a vector of arbitrary inputs in $\mathbb{R}^K$ to a discrete distribution, and as such this function is naturally applied after the final output layer in a neural network and before the CE loss, acting as the final activation.

$$\mathsf{softmax(x)} : \mathbb{R}^K \to [0,1]^K$$
$$\mathsf{softmax}(x)_j = \frac{\exp(x_j)}{\sum_{c=1}^{K} \exp(x_c)}$$

After applying the softmax, the values in the output vector lie in the interval $[0, 1]$ and sum up to 1, thus respecting the two rules introduced above which ensure a valid probability distribution. The numerator, due to the e^{x_j} factor, bounds the input values to be in the positive range, while the summation in the denominator normalizes the vector so that the output values sum to 1. The Cross Entropy loss is then applied right after the softmax. In this formulation $\hat{y}$ and y denote a matrix of size $N \times K$, where N is the number of datapoints.

$$\mathsf{L_{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{K} y_{ic} \log(\mathsf{softmax}(\hat{y_i})_c) = -\frac{1}{N} \sum_{i=1}^{N} \log(\mathsf{softmax}(\hat{y_i})_{l(i)})$$

Since y is 1-of-K encoded, it can be easily derived that both L_{CE} formulas above are identical. We denote the real class as $l(i)$ to indicate the dependency on i.

2.1.6 Backpropagation and Gradient Descent

So far we have explained how the computation is processed in a feedforward network and the purpose of loss functions. A key question still left to answer is how exactly to adjust the network parameters θ to some training data (x, y) with the end goal of performing prediction on some previously unseen inputs

x^*. Backpropagation (Rumelhart et al., 1986) is an algorithm for computing gradients of the loss function $\frac{\partial L}{\partial \theta}$ in respect to network parameters θ. The computation graph is a useful abstraction to motivate backpropagation. Without loss of generality, we may think of a neural network as a series of mathematical operations where each operation receives some inputs and produces some outputs. Even the most complex networks can be decomposed into basic operations like addition or multiplication carried over vector or matrix inputs and outputs. Figure 2.4 illustrates an example operation which receives inputs x and y and produces output z. Importantly, the inputs x and y are actually the outputs from previous operations and after computing f the output z will become input to other upstream operations.

A computation graph is essentially a composition of basic operations where each node represents an operation and each edge represents data flow between operations. For each operation f, we can calculate a local partial derivative $\frac{\partial f}{\partial x_i}$ over each one of the inputs x_i if the operation itself is differentiable. Intuitively, the partial derivative represents the rate of change on the output $f(x_1, .., x_n)$ produced by a slight change over an input x_i. Even though we can calculate local gradients, we seek to understand the global impact of changing parameters θ on the loss function L. The chain rule of calculus allows for gradients to be propagated throughout the computational graph. We present an example for the application of the chain rule on the operation of Figure 2.4. The upstream gradient is multiplied with the local gradient and then fed backwards to previous nodes.

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x} \qquad\qquad \frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$$

The training of neural networks is thus divided into two steps: a forward pass calculates all intermediate operations starting at the network inputs x and ending at the network predictions $\hat{y}$ and loss function L. A backward pass propagates the error signal backwards through the same operations in reverse order. The error in respect to the parameters is calculated, then followed by a parameter update. The process is repeated for a specific number of iterations or until convergence.

A parameter update rule defines how exactly to update the network parameters θ according to the computed gradients $\frac{\partial L}{\partial \theta}$. Gradient descent is a widely known iterative algorithm for optimizing neural networks. After computing the gradients from the backward pass, the gradient descent algorithm takes a step on the parameter space in the opposite direction of the gradient. A learning rate η specifies the extent of the update (i.e., how large a step should be). After enough iterations, the parameters will arrive at a local minimum of the loss function.

$$\theta = \theta - \eta \frac{\partial L}{\partial \theta}$$

Batch (i.e., standard) gradient descent computes the gradients for all training examples, which may be prohibitive for large training sets of thousands or millions of examples. On the other hand, Stochastic

[1] http://cs231n.github.io/

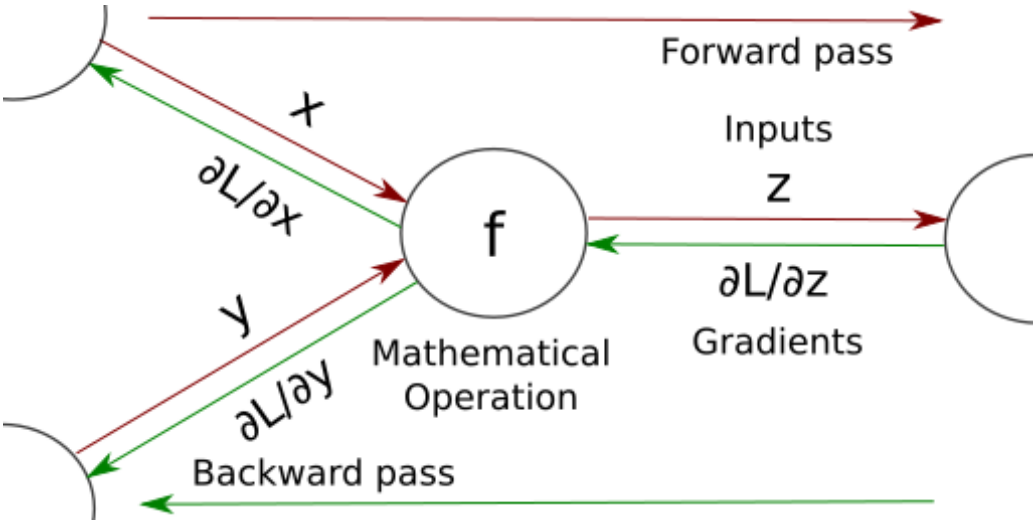

Figure 2.4: Forward and backward passes of a local operation, adapted from the course notes of Stanford's CS231n online course[1].

Gradient Descent (SGD) computes the gradients for a single training example, thus resulting in much faster updates at the expense of incurring noisier updates due to higher variance. A compromise between batch gradient descent and SGD, in terms of lowering the speed of computation and the variance of the parameter updates, corresponds to mini-batch gradient descent. At each training iteration, a mini-batch is sampled from the training data, which corresponds to a fixed-sized subset of the training data. The gradients are then updated after computing the forward and backward passes.

Adaptive Moment Estimation (Adam) (Kingma and Ba, 2014) is a SGD variant which has been broadly adopted for training deep learning models. The update rule of SGD performs gradient updates equally for all parameters via a global learning rate, which is held fixed during training and is commonly hard to tune for achieving good convergence. Adam computes per-parameter adaptive learning rates, meaning that each model parameter benefits from having its own learning rate, which is further adapted on a per-parameter basis as training progresses. Adam stores an exponentially decaying cache of previous gradients. Decaying helps in reducing the influence of gradients computed in the far past. Parameters which receive large updates over time cause the effective learning rate to decrease, while small updates cause the effective learning rate to increase.

2.1.7 Recurrent Neural Networks

One notable limitation of feedforward networks is that the input and output dimensionality is fixed during and after network training, further complicating the task of processing variable length data. Some tasks certainly fit well within the paradigm of fixed length input and output, such as classifying 32×32 images into prespecified digit categories. However, for some other tasks, being confined to fixed dimensionality

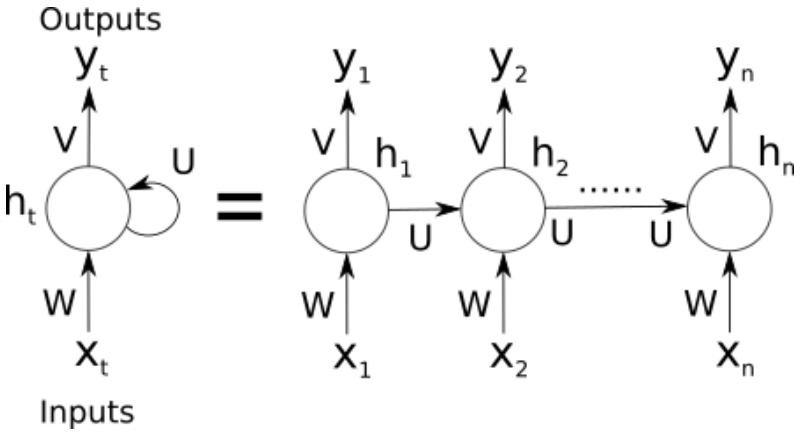

Figure 2.5: RNN architecture (left) and RNN unrolling in time (right).

becomes a major restriction. Recurrent neural networks (RNNs) are a family of neural network models specifically tailored for sequential data. Importantly, the architecture of RNNs enables the model to operate on variable length input and output, thus resulting in a model attractive to problems with an inherent sequential nature (see Figure 2.5). In the case of textual contents, the sequential dimension commonly encodes the position of the linguistic unit (e.g., paragraph, sentence, word, or character) in the analyzed text. In statistical terms, variables at step t are dependent on variables at other steps, and as such the i.i.d. assumption does not hold along the sequence.

Differently from feedforward networks, loops in the computation graph are allowed between recurrent connections (the self-arrow in Figure 2.5). Unrolling the network depicts how the computation can be structured in a feedforward or sequential manner by using vector outputs from the previous step as inputs to the current step. Formally, an RNN takes an input sequence $\mathbf{x}^T = (x_1, x_2, ..., x_n)$ and produces a hidden state sequence $\mathbf{h}^T = (h_1, h_2, ..., h_n)$ and an optional output sequence $\mathbf{y}^T = (y_1, y_2, ..., y_n)$. The hidden state h_t acts as a kind of memory in the network, being capable of remembering past information that can span several steps in the past encoded along the hidden vector dimensions.

The same set of operations are applied at each step as described by an RNN cell. Given an input x_t at step t and a hidden state h_{t-1} from the previous step, an RNN cell describes a set of equations to produce a hidden state h_t at the current step t. Elman RNNs (Elman, 1990) are a simple type of RNN cell commonly found in the literature. The weight matrices W, U and V, together with bias terms b_h and b_y are the parameters of an Elman RNN. These parameters are shared, in the sense that instead of having different sets of parameters dependent on the step t, the same set of parameters are applied in all different steps $t = 1, 2, ..., n$. An Elman RNN is described as:

$$h_t = \sigma(Wx_t + Uh_{t-1} + b_h)$$
$$y_t = \sigma(Vh_t + b_y)$$

Researchers have identified two main issues in training RNNs with long dependencies that can extend to several steps in the past, namely the problems of exploding and vanishing gradients. Vanishing gradients, or its counterpart exploding gradients, occur whenever the computed gradients from backpropagation become either too small to have any reasonable effect on the loss surface, thereby requiring a higher number of gradient updates, or alternatively become too large, thus causing considerable fluctuations on the parameter space and hindering the ability of finding a local optimum. As detailed in Pascanu et al. (2013), gradient clipping is a simple yet effective strategy of dealing with exploding gradients by rescaling the gradients whenever their norm exceeds a certain threshold.

The Elman RNN cell that was just described is rarely used in practice due to being affected by the vanishing gradients problem and, as such, it is not capable of reasonably retaining long term information. Instead, more advanced cell types have been developed with the specific purpose of alleviating the problem of vanishing gradients in modeling long term dependencies. Long Short-Term Memory (LSTMs) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Cho et al., 2014) are popular choices of RNN cells for this purpose.

LSTMs introduce a gating mechanism which can control the flow of values by employing a sigmoid function σ to limit the output range to be between 0 and 1, where 0 means the values are not used at all and 1 means the values are fully used. In addition to the hidden state h_t, LSTM cells feature a cell state c_t which acts as an internal cell memory. Information is either forgotten or added as controlled by the gates. A forget gate f decides which information to keep and which information to discard from previous cell state c_{t-1}, while an input gate i decides on the amount of new information to be added to the cell state c_t. After calculating c_t, an output gate decides which information to output from c_t. Information not contained in the output is kept internally inside the cell state and can be used in subsequent steps. The gates are described as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
g_t &= \tanh(W_g x_t + U_g h_{t-1} + b_g) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

Each gate uses its own set of weight matrices and biases. Additionally, the gates operate by employing a linear combination of the current input x_t and previous hidden state h_{t-1} followed by a sigmoid nonlinearity. The vector g corresponds to candidate new information, whose values will first be filtered by the input gate i before being added to c_t. A $\tanh$ nonlinearity is applied to calculate g and h_t, having the effect of bounding the output values to lie between -1 and 1.

An alternative recent formulation is that of GRUs, which completely do away with the internal cell

state c_t used in LSTMs. In a GRU architecture, the hidden state h_t fulfills the roles of both the exposed output state and the memory component. As a consequence, the output gate, present in LSTMs, does not exist in GRUs since its purpose was to control the amount of information from c_t to transition into h_t. A reset gate r decides on how best to combine x_t and h_{t-1} to create a candidate hidden vector $\tilde{h_t}$, while an update gate z decides how much past information h_{t-1} should be retained and how much new information $\tilde{h_t}$ should be added, effectively merging the forget and input gates in an LSTM.

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\
\tilde{h_t} &= \tanh(W^h x_t + U^h (r \circ h_{t-1})) \\
h_t &= (1 - z) \circ h_{t-1} + z \circ \tilde{h_t}
\end{aligned}
$$

In comparison to LSTM networks, GRUs are faster to train and have less parameters due to the lack of the output gate. Greff et al. (2015) performed extensive evaluation experiments over several RNN variants across multiple tasks, and their results showed that different variants usually lead to similar performances across tasks, hinting that the particular choice of cell type, for instance LSTMs or GRUs, might not be as important for learning when compared to effective tuning of other key hyperparameters like the learning rate or hidden layer size.

Bidirectional RNNs (BiRNNs) are an extension of RNNs which read an input sequence in both directions. BiRNNs are composed of two RNNs, namely a forward RNN that reads the input from left to right, therefore capturing unbounded left side context, and a backward RNN that reads the input from right to left, therefore capturing unbounded right side context. At each step, the hidden states for the forward and backward RNNs are concatenated as $h_t = h_t^f \oplus h_t^b$, where $\oplus$ denotes the concatenation operation. Thus, h_t captures the context at both sides of t.

2.2 Related Work

This section first covers previous work addressing the sequence labeling tasks of POS tagging or NER, highlighting studies which evaluate on these tasks of interest over Portuguese or Spanish texts. We also review multi-task and cross-language learning methods that are directly applicable for these languages.

2.2.1 Neural Approaches for Sequence Labeling

Neural network methods for sequence labeling have enjoyed a great deal of interest in recent years. The early work of Collobert et al. (2011), in turn building upon Collobert and Weston (2008), had a pronounced impact in establishing the usefulness of neural network models for NLP, and in steering the

field away from linear statistical models leveraging hand-coded features (Goldberg, 2017). The authors discarded extensive NLP knowledge in the form of feature engineering in order to create a single model architecture which could be applied on multiple sequence labeling tasks: POS tagging, chunking, NER and semantic role labeling. Word embeddings were learnt from a large corpora of unlabeled data, and a word window model or a Convolutional Neural Network (CNN) (LeCun et al., 1989) were applied on the concatenation of the word representations obtained from a lookup layer for a given sentence. At the output layer, the training criterion was defined as the minimization of the log-likelihood w.r.t the parameters, considering either independent word predictions via a softmax, or interactions between output labels via a Conditional Random Field (CRF) (Lafferty et al., 2001) loss.

Santos and Zadrozny (2014) proposed an approach which extracts word- and character-level information from words, via a combination of word embeddings and a Char-CNN model. Similarly to the work of Collobert et al. (2011), the resulting word representation was used as input to a word window for capturing word context, followed by a CRF for structured inference. The authors achieved state-of-the-art results for POS tagging on the Penn Treebank (PTB) corpus for English, and on the Mac-Morpho corpus for Portuguese, evaluated in terms of tagging accuracy. On follow-up work, Santos and Guimaraes (2015) employed the same architecture without major hyperparameter changes for NER, obtaining state-of-the-art results on the HAREM I corpus for Portuguese, and on the CoNLL-2002 Spanish corpus, in terms of the F1 score at the entity-level. The authors investigated the impact of combining alternate word representations, namely word embeddings, hand-coded features (suffixes and capitalization) and the character-level representations obtained from the Char-CNN model. Their experiments showed that consistent improvements could be accomplished by combining different word representations, as compared to using either representation in isolation.

Chiu and Nichols (2015) proposed an architecture which combines a CNN at the character-level with Bi-directional Long Short-term Memory (BiLSTM) units, to model word context. A BiLSTM captures unbounded left and right side context and is perhaps the main improvement over the window-based model of Santos and Guimaraes (2015). At each time step, the concatenation of the forward and backward representations of each LSTM unit is passed to an affine layer, and a softmax over the output labels computes probabilities for each tag category. The authors evaluated their model on the English CoNLL-2003 and OntoNotes 5.0 corpora with NER annotations, showing the superiority of the combined model and establishing a new state-of-the-art on both datasets by employing a combination of word- and character-level representations with capitalization and lexicon features.

Ling et al. (2015) proposed the Char-BiLSTM model and evaluated it on language modeling and POS tagging. A hierarchical LSTM arrangement is employed for both tasks, albeit with slight modifications. A BiLSTM at the character-level creates a representation of word morphology, which is then fed to an upstream word-level BiLSTM for capturing word context in the case of POS tagging, or a word-level LSTM

for capturing context of past words in the case of language modeling. The results achieved by Ling et al. (2015) suggest that a character-level representation, obtained solely from the characters, consistently delivers better performance than a word-level representation using word embeddings. Over five different languages, the Char-BiLSTM model consistently led to perplexity reductions on language modeling compared to a model relying solely on word lookups. The results were quite similar for POS tagging, as the character-level representation outperformed word embeddings on all tested five languages. Improvements were especially pronounced for morphologically rich languages (i.e., languages with a high number of morphemes per word), as the complex word structure which governs these languages can be somewhat approximated with a character-level model.

Huang et al. (2015) proposed a set of models applicable for sequence labeling tasks based on some variations of LSTMs and CRFs. The different models that were studied by these authors differ in two main aspects: whether a LSTM or a BiLSTM are employed for modeling word context, and whether the output layer employs a softmax or a CRF for label decoding. In order to determine the effectiveness of the proposed models, the authors report results on several benchmark sequence labeling tasks, although only considering the English language: PTB for POS tagging, CoNLL-2000 for chunking, and CoNLL-2003 for named entity recognition. A linear CRF model formed a strong baseline in all datasets, performing especially well at NER. While a LSTM model performed poorly, a BiLSTM model improved upon the LSTM performance by considering context at either side of words, thus approaching the performance of the linear CRF tagger, except at the NER task where the CRF baseline still outperformed the BiLSTM. Across multiple datasets and setups, a LSTM-CRF model always outperformed both the CRF and the BiLSTM baselines, showing the effectiveness of the combined approach. Further improvements were accomplished with a BiLSTM-CRF.

Lample et al. (2016) introduced two model architectures in the context of named entity recognition and achieved state-of-the-art results on standard NER benchmarks in four different languages. One of the architectures employs an approach similar to shift-reduce dependency parsers using Stack-LSTMs (Dyer et al., 2015). Although a useful architecture per se, we shall not explore Stack-LSTMs on this work, mainly due to the fact the other proposed architecture achieved a better performance on the experiments. The other architecture, which is similar to the BiLSTM-CRF model proposed by Huang et al. (2015), uses subword information extracted with a Char-BiLSTM model. The proposed model works as follows: a Char-BiLSTM model creates a character-level representation, which is then concatenated with a word embedding vector for each word in the sequence. The resulting word representation is used as input to a word-level BiLSTM. At the output level, a final CRF layer models correlations between labels, which was found to be important for NER since common tagging schemes enforce hard contraints on the possible label sequences. The proposed architecture unified into a single model past research on learning character-level representations from words (Ling et al., 2015) and jointly decoding the best

	Task	Dataset	Word inputs	Context	Label decoding
Collobert et al. (2011)	POS NER	PTB CoNLL-03	$\overrightarrow{w} + \overrightarrow{f}$	Window/CNN	Softmax/CRF
Santos and Zadrozny (2014)	POS	PTB/Mac-Morpho	$\overrightarrow{w} + \overrightarrow{c_1}$	Window	CRF
Santos and Guimaraes (2015)	NER	CoNLL-02/HAREM	$\overrightarrow{w} + \overrightarrow{c_1}$	Window	CRF
Ling et al. (2015)	POS	PTB/Floresta	$\overrightarrow{w} + \overrightarrow{c_2}$	BiLSTM	Softmax
Huang et al. (2015)	POS NER	PTB CoNLL-03	$\overrightarrow{w} + \overrightarrow{f}$	LSTM/BiLSTM	Softmax/CRF
Chiu and Nichols (2015)	NER	CoNLL-03/OntoNotes	$\overrightarrow{w} + \overrightarrow{f} + \overrightarrow{c_1}$	BiLSTM	Softmax
Lample et al. (2016)	NER	CoNLL-02/03	$\overrightarrow{w} + \overrightarrow{c_2}$	BiLSTM	CRF
Ma and Hovy (2016)	POS NER	PTB CoNLL-03	$\overrightarrow{w} + \overrightarrow{c_1}$	BiLSTM	CRF

Table 2.1: Comparison of past work addressing sequence labeling for NLP. In the column corresponding to word inputs, $\overrightarrow{w}$ refers to word embeddings, $\overrightarrow{f}$ to hand-coded features, $\overrightarrow{c_1}$ to Char-CNN and $\overrightarrow{c_2}$ to Char-BiLSTM.

label sequence based on sentence-level label information (Collobert et al., 2011; Huang et al., 2015). The authors trained their models for English, Spanish, German and Dutch, using the CoNLL-2002 and CoNLL-2003 NER datasets. Their models achieved significant improvements over previously reported results for German and Spanish, and surpassed all except one previous work for English and Dutch. Dropout (Srivastava et al., 2014) regularization was found to be crucial for obtaining good results, and dropout masks were inserted after concatenating the word and character representations, and before the input to the BiLSTM, thus forcing the model to rely on both input representations. Although the authors have evaluated the proposed model solely on NER, the general nature of the architecture translates into an effective model for other sequence labeling tasks and/or domains.

Ma and Hovy (2016) proposed an end-to-end architecture for sequence labeling which obtained state-of-the-art results on POS tagging and NER. The architecture is in all aspects identical to the one proposed by Lample et al. (2016), except that word morphology is encoded via the Char-CNN model instead of the Char-BiLSTM model. Previous work (Santos and Zadrozny, 2014; Chiu and Nichols, 2015) had confirmed the usefulness of the CNN-derived character representation in extracting morphological features from words. Furthermore, the preliminary results of Chiu and Nichols (2015) had shown that a Char-BiLSTM model did not improve upon the Char-CNN model, while being noticeably slower to train. The remainder parts of the model are identical to past work: the word representation is a concatenation of the vectors from the word embedding and the Char-CNN, and a dropout layer is applied after the concatenation. A BiLSTM creates a context-dependent representation for each input word and a CRF layer models label correlations. Differently from previous approaches, the authors do not perform any kind of task-specific data preprocessing or feature engineering, and still achieve state-of-the-art results with their task agnostic model. The model scored 97.55 tagging accuracy on the PTB corpus, and

a F1 score of 91.21 on the English CoNLL-2003 NER task, which is an improvement of 0.01 over the previously reported best result employing feature engineering and external resources. Table 2.1 provides a summary of the approaches for sequence labeling described in this section, together with the evaluation tasks and datasets from previous work.

2.2.2 Multi-Task and Cross-language Learning for Sequence Labeling

Approaches for multi-task learning train a single model to solve multiple tasks through parameter sharing. First proposed in the 1990s (Caruana, 1997), this idea was first applied with neural networks in the work of Collobert and Weston (2008), where the authors share the embedding matrices across tasks.

Søgaard and Goldberg (2016) suggested that instead of predicting all tasks at the same output layer in a multi-task scenario, task supervision should come from different RNN levels. They associated an RNN level with each task (e.g., a lower level predicted POS tags, while a higher level predicted chunking tags). This model effectively combines multi-task and cascaded learning, exploiting a natural order among NLP tasks from which high-level tasks should benefit more from lower-level tasks. The proposed model led to improvements only if the tasks were fairly similar (e.g., same syntactic nature), as the authors reported no improvements for NER.

Hashimoto et al. (2016) extended this line of work by proposing a deep architecture that solves many different NLP tasks at various successive layers in an end-to-end paradigm. The complexity of the tasks increases with the layer levels, with more complex tasks being solved at the upper levels. The first two layers solve word-level syntactic tasks. More concretely, the first layer solves POS tagging, while the second layer solves chunking. A BiLSTM network with a softmax at the output layer produces a sequence of POS tags from the word inputs. Another BiLSTM network at the second level receives as input the sequence of words, the sequence of POS tags which was predicted at the first BiLSTM layer, and the sequence of hidden states of the first BiLSTM layer. A sequence of chunking tags is then predicted, which takes into account all this lower-level information. A third BiLSTM layer receives information from the previous two layers in order to solve dependency parsing. Two semantic tasks, namely semantic relatedness and textual entailment, operate at the topmost layers, and as such benefit from a great deal of lower-level information. Both of these tasks receive as input two sentences, which were first passed through the initial three layers, and employ an encoder architecture to derive a sentence-level rather than a word-level representation. A score with a task-specific interpretation is then computed from the sentence-level representation after the encoding of both sentences.

Plank et al. (2016) evaluated LSTM-based models for POS tagging over a large number of languages and across different input representations. The work employs treebank data for 22 languages obtained from the Universal Dependencies project (Nivre et al., 2016). The authors built a POS tagger using a BiLSTM for capturing word context, and evaluated the usefulness of different input representations

	Task	Dataset	Task supervision	Extra objective
Søgaard and Goldberg (2016)	POS Chunking CGG supertagging	PTB	Cascaded	-
Hashimoto et al. (2016)	POS Chunking Dependency parsing Semantic relatedness Textual entailment	PTB PTB PTB SICK SICK	Cascaded	-
Plank et al. (2016)	POS coarse-grained	22 languages UD	Outermost	Word frequency
Horsmann and Zesch (2017)	POS fine-grained	22 languages	Outermost	Word frequency
Yang et al. (2017)	POS Chunking NER	PTB CoNLL-00 CoNLL-03	Task-wise CRF layer	-
Rei (2017)	POS Chunking NER	PTB/UD-ES/FI CoNLL-00 CoNLL-03	Outermost	Language modeling

Table 2.2: Comparison of past work on transfer learning approaches for sequence labeling.

for each language. More concretely, the input representation to the word-level BiLSTM may be one of: word embeddings with random initialization $\overrightarrow{w}$, character-level representations based on character embeddings $\overrightarrow{c}$ obtained from applying the Char-BiLSTM model, or the combination of these representations. Whenever multiple representations were in use, the authors concatenated each individual representation in order to generate a single word representation which was then fed to the upstream word-level BiLSTM.

The results of their multilingual study show that a BiLSTM tagger using inputs $\overrightarrow{w}$ is easily surpassed by a baseline implementation of an HMM tagger on all but three languages, suggesting that deep learning models without considerable efforts devoted to architecture engineering are not an outright replacement for well optimized traditional taggers. A BiLSTM tagger using inputs $\overrightarrow{c}$ performed reasonably well and even surpassed the HMM tagger on nine languages, hinting that the character-level representations are an important architectural component for obtaining good accuracy at POS tagging, and that relying on characters alone already provides reasonable accuracy in comparison with taggers which rely heavily on feature engineering. As these representations are somewhat effective in their own right, the best representation $\overrightarrow{w} + \overrightarrow{c}$ comes from combining word- and character-level representations, with the combined representation surpassing the HMM baseline on all but one language. Intuitively, each of these representations captures different linguistic phenomena: representations at the word-level capture syntactic and semantic information, while representations at the character-level capture orthographic information. The $\overrightarrow{w} + \overrightarrow{c}$ representation with pre-trained word embeddings led to consistent accuracy improvements compared to training the word embedding matrix from random initialization, as supported by the findings from Collobert et al. (2011).

Most improvements over the baseline actually came from predicting tags for rare words, as the HMM

model struggles to predict tags for words which were seen few times on the training corpora. With that in mind, the authors also proposed a multi-task learning objective as an addition to the POS loss function. At each step, besides predicting the POS tag for the current word, the model will also learn to predict the log frequency of the current word. The expectation is that the underlying shared representation will become predictive of word frequency and thus handle differently regular and rare words. With this addition, the authors consistently improve the out-of-vocabulary accuracy rates. The log frequency objective can be interpreted as an instance of multi-task learning, forcing the model to be predictive for two tasks at the same time. The shared representation must become informative enough for both tasks, in order to solve them jointly. Deep learning models are known for requiring large datasets in order to generalize well, and the authors also investigated the impact in model accuracy of varying dataset sizes, concluding that to achieve good performance the best deep learning model actually required less data than previously thought.

Horsmann and Zesch (2017) investigated whether the models described by Plank et al. (2016) could be extended to operate on fine-grained tagsets. Plank et al. (2016) used solely treebanks available on the Universal Dependencies project, which are standardly annotated with coarse-grained tagsets containing a maximum of 17 POS tags. For morphologically rich languages, coarse-grained tagsets are only able to capture a fraction of the information contained in word structure, disregarding important morphological clues such as case or gender. The authors therefore assembled a set of 27 treebanks of fine-grained tagsets over 21 languages. The LSTM model of Plank et al. (2016) was used with the same configurations, and similarly the authors investigated the impact on accuracy over different inputs: Word $\overrightarrow{w}$, Char $\overrightarrow{c}$, Word-Char $\overrightarrow{w} + \overrightarrow{c}$ and Word-Char+, which is the Word-Char model with the multi-task objective. All these input representations are computed the same way as described previously. The Char representation $\overrightarrow{c}$ provided accuracy on par with traditional taggers. For morphological rich languages, like Spanish or Hungarian, the Char $\overrightarrow{c}$ inputs outperformed Word $\overrightarrow{w}$ significantly, although for Slavic languages, which used much finer tagsets, the improvements were not so noticeable. Across all languages, Word-Char+ inputs yielded the best results, closely followed by the Word-Char architecture. The main benefit of Word-Char+ over Word-Char comes in the form of out-of-vocabulary accuracy rates, which experience a significant accuracy boost.

Yang et al. (2016, 2017) explored whether general-purpose sequence labeling architectures could learn across tasks and languages. Specifically, the authors employed a model architecture similar to that of Lample et al. (2016), albeit with a hierarchical BiGRU instead of a hierarchical BiLSTM. The proposed architectures use parameter sharing to transfer knowledge between tasks. The type of transfer learning defines the extent of parameter sharing in the network: for cross-domain transfer the authors share all model parameters if the label sets are identical, or if all the label sets can be mapped into a single label set. A cross-domain example with joint training would be POS tagging for the domains

of modern and historical texts. If the labels sets are different, and cannot be mapped into the same label set, then the authors share model parameters for the word and character embeddings and for the word- and character-level networks, but not for the CRF layer, which was allowed to be tailored for each task. This architecture with different CRF layers per task is also used for cross-application (i.e., multi-task) transfer. A multi-task example would be POS tagging and NER for the same language. Finally, a cross-lingual architecture shares only the character embeddings and character-level network, with the remaining word-level and CRF layer components being tailored for each language. This type of parameter sharing implies that the languages must be morphologically similar. The authors achieved promising results on the multi-task setting for 3 tasks: PTB corpus (POS), CoNLL-03 (NER) and CoNLL-00 (chunking), and on the cross-lingual setting for NER on 3 languages: CoNLL-02 (Dutch), CoNLL-02 (Spanish), CoNLL-03 (English). The authors claim state-of-the-art results on all corpora except the PTB corpus.

Rei (2017) proposed a secondary language modeling objective for sequence labeling which requires no additional annotated data. In both the tasks of sequence labeling and language modeling, the inputs are a word sequence, with words being first mapped to a distributed representation and then fed into some RNN variant for capturing word context. Where these tasks differ the most is on the output sequences, which correspond to predicting the adjacent words in the input sequence for language modeling, or a label for the current word in the case of sequence labeling. The authors theorized that the language modeling objective encourages the network to learn a richer representation which may be useful for sequence labeling. At each step, a forward LSTM predicts the next word in the input sequence, while a backward LSTM predicts the previous word in the input sequence. In addition, the combined representation of the forward and backward LSTMs is used to predict a label for the current word. Without extra annotated data, the multi-task objective led to improvements over baseline approaches on the PTB corpus for POS tagging, and on the CoNLL-03 corpus for NER. An extra language modeling objective can be seen as a precursor to the idea of contextual word vectors. Recent approaches such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) incorporate context-dependent word representations from pre-trained language models. In contrast to more traditional non-contextual word embeddings, these approaches model linguistic context and they can handle the existence of many meanings for words (polysemy), having recently shown significant improvements over the state-of-the-art across a wide range of NLP tasks. Table 2.2 shows a summary of the methods and evaluation tasks/datasets described in this section.

3

Joint NER and POS Tagging for Portuguese and Spanish Texts

Contents

The architecture of the neural network model used in our experiments takes inspiration from the previous models proposed by Lample et al. (2016) and Ma and Hovy (2016). Figure 3.1 shows a high-level overview on the proposed approach. For each word, we concatenate the word-level representation provided by pre-trained cross-lingual FastText (Bojanowski et al., 2017; Zhou et al., 2019) embeddings with the character-level representation provided by a BiLSTM over the characters. The resulting word representation is used as input to a word-level BiLSTM network, which captures word context at the left and right side of each word. While the work of Lample et al. (2016) and Ma and Hovy (2016) employs a single CRF layer over the output labels, our approach employs two CRF output layers, one for each task of interest, and in that regard it is similar to the cross-application architecture of Yang et al. (2017). Since some of the datasets contain labels only for one of the two tasks (i.e., either POS or NER tags), we employ a masked loss function which assigns a CRF loss of zero when the ground-truth labels are not available for a dataset. At inference time, our network predicts POS and NER labels for each dataset, even when evaluating results over the datasets containing only the POS or the NER task labels. We follow the hyperparameter recommendations of Reimers and Gurevych (2017), and we also considered a penalized hyperbolic tangent activation within the BiLSTM nodes, as suggested by Eger et al. (2018). The following section details the model architecture, while Section 3.2 details the approach followed for projecting the Portuguese and Spanish word embeddings to a common space. Finally, section 3.3 describes the datasets used in our experiments.

3.1 The Proposed Neural Model

Formally, a dataset for structured (i.e., sequential) prediction is composed of sequences of word representations $x^i = (x_1, x_2, ..., x_L)$ and labels $y^i = (y_1, y_2, ..., y_L)$. Each input x_t corresponds the word at position t. We assume a set of sequences of word-label pairs S are given in the form $(x^i, y^i)_{i=1}^N$, where N is the total number of sequences and L is the length of a word-label sequence. We place no restriction specifying that all sequences must have equal length (i.e., sequence length L may actually vary from sequence to sequence). The training problem can be stated as taking the training samples S and adjusting model parameters θ in order to fit a model m to the underlying data distribution. The prediction problem can be stated as coming up with the most likely label sequence y^* given a novel word sequence x^* and a trained model m. The problem is particularly challenging since the model must naturally take into account the sequential structure of the data.

State-of-the-art systems for sequence labeling create word representations from the provided inputs by combining different information sources. A common approach found in the literature concatenates pre-trained word embeddings, which encode syntactic and semantic information, with character-level representations obtained from Char-BiLSTM models, thus encoding word morphology. Whenever mul-

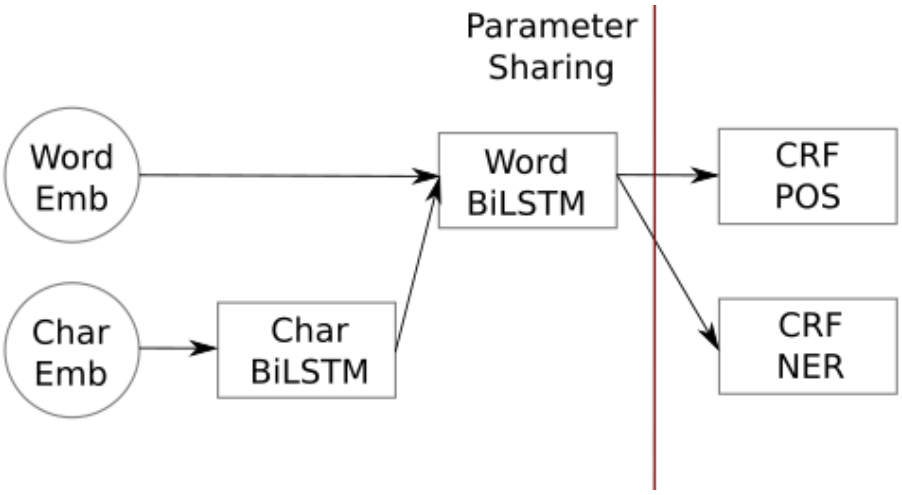

Figure 3.1: Overview on the neural architecture associated to the proposed approach.

tiple representations are in use, these are combined into a single representation to be fed to upstream layers. The concatenation operation is commonly applied for this purpose, with the advantage of preserving the representational power of each individual representation. Each representation encodes some degree of nonoverlapping information, and thus we can benefit from aligning their individual strengths.

Our neural architecture for addressing the aforementioned task combines pre-trained cross-language word embeddings, described in Section 3.2, with character-based representations, leveraging the Char-BiLSTM character compositional model proposed by Ling et al. (2015). The Char-BiLSTM model applies a BiLSTM at the character-level to create a fixed-size representation of word morphology. The model works as follows: given a certain word w, we take the characters $c_1, c_2, .., c_n$ which compose word w, where n is the word length, and obtain character embeddings $E(c_1), E(c_2), .., E(c_n)$ for each character by performing row look-up operations on a randomly initialized character embeddings matrix E. These embeddings are expected to encode character similarity (e.g., vowel *vs.* consonant or uppercase *vs.* lowercase). The parameters for the character embeddings matrix are trained jointly with the rest of the model parameters.

The character embeddings are used as input to a BiLSTM, where the forward LSTM reads the word one character at a time from start to end, and the backward LSTM reads the word one character at a time from the end to the beginning. The hidden state at the last step for the forward LSTM is concatenated with the hidden state at the initial step for the backward LSTM, generating a representation which captures word morphology to be fed in upstream layers. The BiLSTMs read w in both directions, thus maintaining a fresh memory of the word boundaries at either side of w. We denote the fixed-size character-level representation as R_c. We have that LSTM$_f$ and LSTM$_b$ denote a forward and backward LSTM component, respectively, and h_t^f denotes the hidden state as created by the forward LSTM at step

t. Then we can write the character-level representation R_c as:

$$R_c = h_n^f \oplus h_1^b = \text{LSTM}_f(E(c_n), h_{n-1}^f) \oplus \text{LSTM}_b(E(c_1), h_2^b) \tag{3.1}$$

The input sequences of word representations, resulting from the concatenation of R_c with pre-trained FastText embeddings, are afterwards used by separate recurrent neural network layers which attempt to model word context. In human languages, context helps a human listener to disambiguate between alternate meanings. For instance, words such as *address*, *place* or *report* can act as either nouns or verbs on a sentence, depending on context, and a POS tagger that looks at single words in isolation would fail to distinguish between alternate meanings if the word is not taken in context with neighboring words. A BiLSTM can again receive an input sequence $(x_1, x_2, ..., x_L)$ and produce a hidden state sequence $(h_1, h_2, ..., h_L)$ that considers the word context for every word in the input.

More formally, LSTMs (Hochreiter and Schmidhuber, 1997) incorporate a cell state c_t which acts as an internal cell memory for the information at position t, and use a gating mechanism that can control the amount of information from c_t to forget and to retain for subsequent steps. A LSTM cell defines a function LSTM$(x_t,\ h_{t-1})$ which computes a hidden state at position t from the input vector at the current position t and the hidden state from the previous position $t-1$. The equations to compute the hidden state h_t for a LSTM cell are as follows:

$$\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
g_t &= \tanh(W_g x_t + U_g h_{t-1} + b_g) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{3.2}$$

In traditional LSTMs, σ is the element-wise sigmoid function, and $\odot$ is the element-wise product. One particularity of our LSTM cell implementation is that we replace the sigmoid and tanh activations on all gates with the penalized hyperbolic tangent activation function, since this approach consistently led to improvements across multiple NLP tasks and setups (Eger et al., 2018). The penalized hyperbolic tangent can be computed as follows:

$$\text{pentanh}(x) = \begin{cases} \tanh(x) & \text{if } x > 0, \\ 0.25 \times \tanh(x) & \text{if } x \leq 0 \end{cases} \tag{3.3}$$

An LSTM (or a BiLSTM) creates a contextual representation of dimensionality $L \times H$ (or $L \times 2H$ in the case of a BiLSTM) capturing each of the L words in the sequence, where H is the hidden layer size.

Then, a feedforward layer with shared weights over time can be employed to convert the representation to a matrix E of dimension $L \times T$, defined over the sequence length L and tagset size T. Instead of making independent tagging decisions for each output label y_t via a softmax, it is beneficial to model correlations between labels. We use a Conditional Random Field (CRF) to model the sequence of labels jointly, in a way that previous tagging decisions about neighboring labels are taken into account. We define the score S of a sequence (x, y) as the sum of emission scores E output by the BiLSTM network, and learned transition scores P with dimension $T \times T$, which define the score of a transition from label i to label j. We can thus write the score S as:

$$\mathrm{S}(x, y) = \sum_{i=1}^{L} E_{iy_i} + P_{y_{i-1}y_i} \tag{3.4}$$

While being trained, the CRF layer will learn useful constraints from the training data, including hard constraints regarding sequences of labels that should never occur, this way greatly reducing the search space of possible sequences by discouraging invalid sequences, and thus forcing the model to output a valid label sequence. We compute a conditional probability $\Pr(y|x)$ for a label sequence y via a softmax over all possible label sequences.

$$\Pr(y|x) = \frac{\exp(\mathrm{S}(x, y))}{\sum_{y' \in \hat{y}} \exp(\mathrm{S}(x, y'))} \tag{3.5}$$

The model is trained to maximize the log-likelihood function $\sum_{i=1}^{N} \log \Pr(y^i|x^i)$ on a training set. Intuitively, for a given x, the model learns to assign a higher score for the real label sequence y in comparison to all other label sequences $\hat{y}$. As the CRF layer only considers bigram interactions between labels, dynamic programming can be applied for efficient training and inference (Rabiner, 1989).

3.2 Cross-Language Word Embeddings

The choice of word embeddings plays a significant role in model performance. As the majority of word embeddings are trained on modern corpora, using off-the-shelf word embeddings directly would be a poor choice on the understudied domain of historical texts, due to differences in spelling and word usage. As a case in point, an official orthography for Portuguese was only adopted in the early 20th century. We indeed observed a substantial number of Out-of-Vocabulary (OOV) words on historical datasets, especially if the ortography was not standardized to modern spelling conventions. To somewhat alleviate the OOV problem, we used pre-trained FastText word embeddings (Bojanowski et al., 2017), which features subword information as an extension to the Skip-gram model (Mikolov et al., 2013). FastText represents words as a sum of character n-gram vectors, thus allowing the model to build word vectors for

unknown words. In our work, we employed FastText to create word vectors for all words in the evaluation corpora, in this way bypassing the OOV problem found in historical corpora. Our preliminary results confirmed the gains in terms of model performance obtained from building representations of rare or unknown words, when compared to off-the-shelf FastText embeddings.

Cross-language learning approaches enable joint training of multiple languages within a single model. One common approach found in the literature employs a bilingual dictionary (Ruder et al., 2017), which maps words from a source language to a target language. The dictionary approach leverages pre-trained monolingual word embeddings to learn a mapping between the monolingual embedding spaces via a bilingual dictionary, which provides a supervised signal that guides the translation pairs to have a small Euclidean distance in the shared space. We employed the cross-lingual approach by Zhou et al. (2019), which instead of regarding word embeddings as a discrete set of points, this approach models the embedding space as a probability density function, in order to deal with the uncertainty in learning word embeddings. The monolingual embedding spaces of the source and target languages are defined by a Gaussian mixture model with N_s and N_t components, where N_s and N_t are the number of pre-trained word embeddings in the source and target languages respectively. Each Gaussian component is mean-centered at the word embedding, weighted by the unigram distribution $P(s_i)$ of word occurrences.

$$P(S) = \sum_{i=1}^{N_s} P(s_i)\, \mathcal{N}(S|s_i, \sigma_s^2 \mathbf{I}) \tag{3.6}$$

In this formulation, $P(S)$ is the density of all points in the source embedding space and $\mathcal{N}$ is a multivariate isotropic Gaussian. This approach then learns a linear mapping in both directions that better matches the probability densities of both monolingual spaces. After training the bilingual mappings, we had to merge the Portuguese and Spanish embeddings, now aligned in a single vector space, into a single embeddings file due to our interleaved training approach of multiple datasets and languages. Whenever a word w is present in both languages, we combined the embeddings via a weighted average using the unigram distribution as weights.

$$E(w) = \frac{P_s(w)E_s(w) + P_t(w)E_t(w)}{P_s(w) + P_t(w)} \tag{3.7}$$

3.3 The Annotated Datasets

In the process of gathering datasets for our experiments, we surveyed the literature in search of Portuguese and Spanish corpora with POS and/or NER annotations, containing either modern or historical texts from multiple domains. We only considered datasets that are openly available for research purposes, and we now make our list publicly available for further experimentation. Table 3.1 presents an overview of the gathered corpora. As the datasets were created in various research projects, they dif-

fer in terms of the content gathering process, tokenization strategies, tagset formats, and annotation guidelines. Thus, an extensive normalization process was required to ensure a consistent format for all datasets. One goal of our work was to standardize the data format and the annotation schemes across all surveyed corpora, and in order to achieve that goal we had to make decisions regarding the output format. We chose the CoNLL format (Tjong Kim Sang and De Meulder, 2003), since it is widely known in the research community. The CoNLL format contains one token per line, where empty lines denote sentence boundaries. Each line contains a fixed number of fields separated by whitespace, which in our work are the raw token, and the POS/NER tags depending on the available annotations.

Regarding the POS annotations, a logical choice for standardization was the Universal POS (UPOS) tagset from the Universal Dependencies (UD) project. In brief, the Universal Dependencies project (Nivre et al., 2016) is an ongoing effort to develop consistent annotation schemes across languages for treebank data, featuring a coarse-grained tagset of 17 UPOS tags[1]. In our work, we made the following modifications: we did not consider PART tags, as these can typically be replaced by other tags. Since Portuguese relies heavily on contractions, we added a set of new tags for these cases: ADP+DET, ADP+PRON, ADP+ADV and VERB+PRON. When given a choice to separate the contractions into multiple words, or keep these as a single token, we chose the latter and concatenate the POS tags of the individual words (e.g., *por* + *a* $\rightarrow$ *pela*). We applied the same reasoning to clitics to keep these as a single token (e.g., keeping words like *tornou-se* or *fechá-la*). The main motivation to follow this approach is to facilitate the processing of new texts directly, in the form in which they are made available (i.e., to avoid complex tokenization rules and pre-processing texts heavily, when applying a trained model to previously unseen data). During the normalization step, some dataset-specific hyphenation rules were reversed, and the original POS tagsets in each dataset were converted automatically via conversion tables and other dataset-specific rules to the standardized UD tagset.

Regarding the NER annotations, we chose to only keep person (PER), organization (ORG) and location (LOC) names, as these can be found on all corpora that were considered. There were other interesting tags to consider, including time expressions and miscellaneous (MISC) tags, but only a fraction of the datasets had these annotations and their usage is often not consistent among the different corpora. We converted all corpora with NER annotations to the BIO tagging scheme, a commonly used format to represent named entities that can span multiple tokens. Previous work (Reimers and Gurevych, 2017) found that for a modern BiLSTM-based sequence learner, a more expressive IOBES scheme does not bring advantages over a simpler BIO scheme, thus justifying our choice of the BIO encoding of chunks as individual tags.

We complied with the original data splits into train, development and test sections whenever the corpus creators provide splits of the data. For datasets without pre-made splits, we created our data splits

[1] http://universaldependencies.org/u/pos/

Dataset	Language	Task
Bosque (Rademaker et al., 2017)	PT	POS
CINTIL (Barreto et al., 2006)	PT	POS/NER
CIPM (Xavier et al., 1994)	PT	POS
Colonia (Zampieri and Becker, 2013)	PT	POS
GSD (McDonald et al., 2013)	PT	POS
Mac-Morpho (Fonseca et al., 2015)	PT	POS
Paramopama (Mendonça et al., 2015)	PT	NER
Tycho Brahe	PT	POS
Post Scriptum	PT/ES	POS
WikiNER (Nothman et al., 2012)	PT/ES	NER
AnCora (Taulé et al., 2008)	ES	POS/NER
CoNLL-02 (Tjong Kim Sang, 2002)	ES	POS/NER
Relaciones Geográficas	ES	NER

Table 3.1: Datasets used to support the evaluation experiments.

by randomizing the sentence ordering, using a fixed seed for reproducibility, and splitting on 75%, 10% and 15% of the sentences for the train, development and test sections respectively. Table 3.2 provides statistics on the datasets used in our experiments, in terms of sentences, tokens and vocabulary size. Figure 3.2 showcases the relative proportions of the POS tags on each dataset, while Figure 3.3 shows the same information for the NER tags.

In the current version 2.3, the UD project contains about 130 treebanks released in 76 languages. The UD project integrates multiple datasets in Portuguese, most notably the Bosque corpus and the GSD corpus. The Bosque corpus (Rademaker et al., 2017) is part of the larger Floresta Sintática treebank, and contains newswire text in European Portuguese, from CETEMPúblico, and Brazilian Portuguese, from CETENFolha. Bosque was fully revised by linguists and conversion rules were applied to convert the original data into the UD format. The GSD corpus of Brazilian Portuguese (McDonald et al., 2013) corresponds to annotated samples from news and blogs, converted from the legacy Google Universal Dependency Treebank. The UD project also features datasets in Spanish, most notably the AnCora corpus. For the reason detailed further ahead, we considered a different version of AnCora instead of the AnCora corpus available on the UD project.

We extracted all the multi-word tokens from CoNLL-U[2] files associated to UD project datasets, and discarded individual words that are included in multi-word tokens. For Portuguese and Spanish, this means that we keep the contracted form of words rather than their uncontracted forms. Though we leave the Bosque corpus mostly unchanged, we heavily preprocessed the GSD corpus. We reverted

[2]http://universaldependencies.org/format.html

Language	Dataset	Task	Sentences	Tokens	Vocab. size
Portuguese	Bosque	POS	9366	210962	28824
	CINTIL	POS/NER	30344	638746	50417
	CIPM	POS	6939	82017	10661
	Colonia	POS	217338	5559844	182629
	GSD	POS	12080	293052	34582
	Mac-Morpho	POS	49932	943775	61938
	Paramopama	NER	16276	386799	38529
	Post Scriptum	POS	36429	813846	68903
	Tycho Brahe	POS	105450	2032394	98111
	WikiNER	NER	142112	3499683	116084
Spanish	AnCora	POS/NER	12146	394485	33943
	CoNLL-2002	POS/NER	11755	369171	31405
	Post Scriptum	POS	29562	671080	52184
	Relaciones Geográficas	NER	917	61583	6403
	WikiNER	NER	127036	3499998	116233

Table 3.2: Statistical characterization for the datasets used in our experiments.

the hyphenation rules from the GSD corpus, which adopts a tokenization strategy of separating all compound words (e.g., *ex - presidente* is represented by 3 tokens). To perform this, we created rules to convert compound words to a single token. These compound words were then manually checked for correctness, and a list of exceptions was created for cases where the tokenization was not being performed correctly. We also revised tokens that were assigned to the incorrect POS tag category, such as contraction words, symbols and URLs tagged as X (other).

CINTIL (Barreto et al., 2006) is a corpus of European Portuguese developed at the University of Lisbon. It contains texts from written sources (e.g., newswire and literature) and spoken sources (e.g., telephone transcriptions, public/private conversations), annotated for named entities and POS tags. CINTIL features a fine-grained POS tagset, containing inflectional features for verbs (e.g., tense, mood, person, number) and names (e.g., gender, number). In addition, the authors chose to separate common contractions of the Portuguese language into multiple tokens (e.g., *pela* $\rightarrow$ *por*$_{\_}$ + *a*, *num* $\rightarrow$ *em*$_{\_}$ + *um*), with a leading underscore to indicate the starting token of the contraction. In our conversion, we curated a list of contractions and their uncontracted forms for the Portuguese language, and used this list to convert contractions to the single token form. CINTIL features some Multi-Word Expression (MWE) tags (i.e., locuções, in the Portuguese language), meaning that the same MWE tag is assigned to a chunk of tokens. As UD annotates tokens at their individual level, we employed a data-driven approach with manual corrections to create conversion rules for MWE tags to UD tags. Regarding NER, we ignore WRK (work) entities and MSC (miscellaneous) entities, keeping only person, organization and location names.

The Mac-Morpho corpus (Aluísio et al., 2003), later revised by Fonseca et al. (2015), was developed by the NILC group at the University of São Paulo and contains newswire text annotated with POS tags in

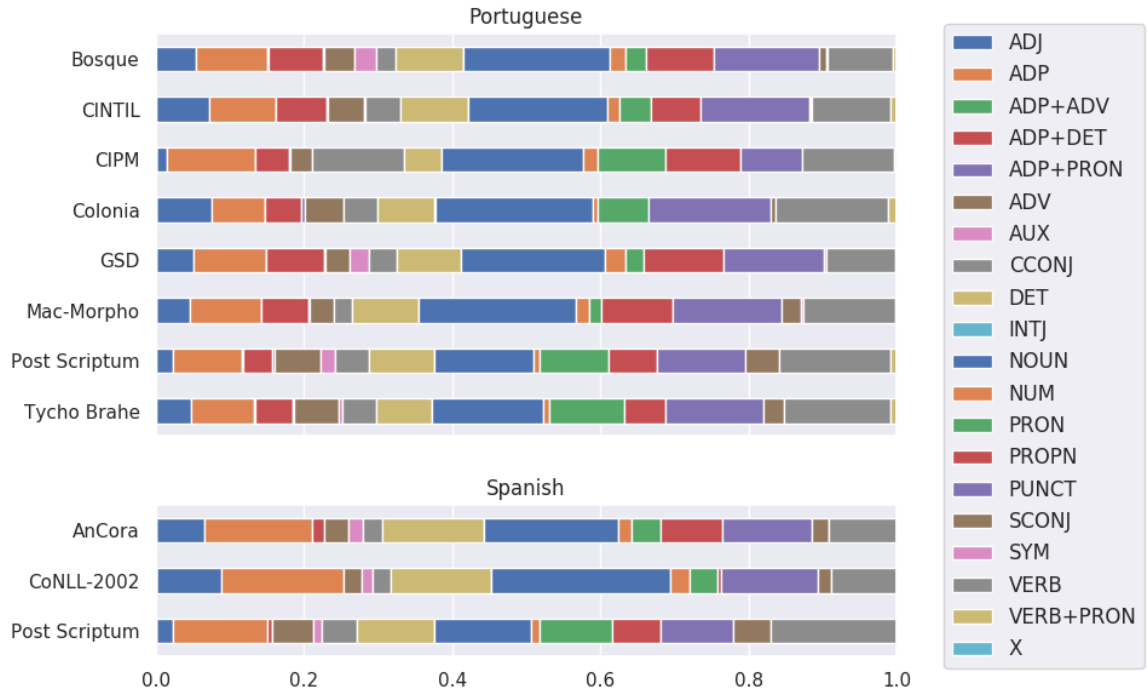

Figure 3.2: Relative proportions of POS tag categories per dataset.

Brazilian Portuguese. The sentences were extracted from the newspaper Folha de São Paulo and cover a wide range of topics (e.g., agriculture, politics or sports). Two main revisions to the corpus were taken since the first release in 2003, and in this work we use the latest revision as described by Fonseca et al. (2015). While the original corpus separated contractions, similarly to the CINTIL corpus, the revised corpus rejoined these tokens into a single token as multi-word contractions, while at the same time concatenating the POS tags of the individual words (e.g., the word *do* is tagged as PREP+ART). We build upon the work of Freitas et al. (2018) that produced conversion rules for Mac-Morpho v1 to UD v1. In our work, however, we only use a subset of their rules[3], since we convert from the Mac-Morpho v3 tagset to the UD v2 tagset. These latest versions have slight tagset modifications, compared to the original versions which was the focus of their tagset conversion efforts.

The Colonia corpus of historical Portuguese (Zampieri and Becker, 2013) consists of texts written between 1500 to 1936. The corpus contains a balanced variety of 48 European Portuguese and 52 Brazilian Portuguese texts. Word lemmas and POS tags were generated with TreeTagger, a probabilistic POS tagger reported to achieve accuracy higher than 95%. As POS taggers are most commonly trained on contemporary data, the authors had to perform a post-processing step to correct unknown lemmas. However, to the best of our knowledge there was no extensive manual revision of the POS tags. At the contraction-level, the authors use compound tags for prepositions with determiner/pronouns and

[3]http://github.com/own-pt/macmorpho-UD

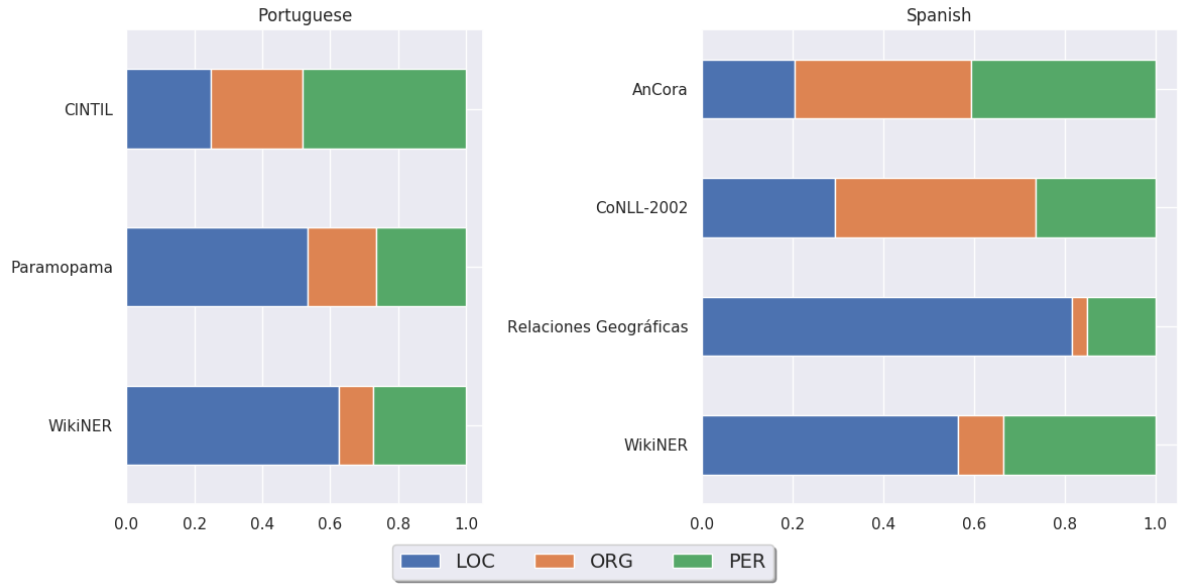

Figure 3.3: Relative proportions of NER tag categories per dataset.

verbs with clitics (enclisis). While this is a fairly large corpus, there is some degree of overlap with other historical corpora, such as the Tycho Brahe corpus. The fact that the POS annotations were automatically generated with no human corrections at a post-processing phase also makes us question the corpus quality. For our purposes, we chose not to train our cross-domain, multi-task and cross-lingual models on this dataset, but we still include it for evaluation in order to determine whether the models generalize to unseen data on the same historical domain.

The Tycho Brahe corpus[4] of historical Portuguese consists of texts written by Portuguese authors born between 1380 and 1881. The corpus contains 76 texts with a total of 3.3M tokens freely available for research. A subset of these texts contain annotations for POS tags, which at the time of writing are 47 texts with a total of 2.0M tokens. We concatenated all texts with available POS tags, and removed metadata in the form of XML tags. The Tycho Brahe corpus separates all forms of Portuguese contractions, with special syntax to mark contraction words (i.e., words that belong to a contraction). We employ rules to join words that belong to a contraction into a single token, and concatenate the POS tags in doing so. The corpus uses a fine-grained POS tagset, containing for instance information about gender and number for nominal tags, and mood and tense for verbal tags. Each verb that can take the role of auxiliary verb in Portuguese has its own POS tag. We simplify the tagset by removing this type of extra information, and in that way condensing it to the coarse-grained UD tagset.

Corpus Informatizado do Português Medieval (CIPM) (Xavier et al., 1994) is a historical Portuguese corpus developed at Universidade Nova de Lisboa, containing early Portuguese texts from the 12th to

[4] http://www.tycho.iel.unicamp.br/corpus/

the 16th centuries. CIPM includes literary texts (e.g., travel narratives and doctrinal prose), and non-literary texts primarily of legal nature (e.g., private notarial documents and royal documents). We wrote a small web crawler to fetch all available texts with POS annotations, and we heavily pre-processed the texts. We removed all sorts of metadata inside the texts, such as XML section meta tags, comments with meta information and non-annotated excerpts in Latin. The digitization and/or transcription of historical texts is a complex process, and due to the nature of this process the authors insert special syntax within the text. Some of this extra syntax signals word abbreviations, character or word corrections by the authors, unknown segments due to the deterioration of the source material, deliberate suppression of text fragments by the authors, etc. We created several rules to remove such syntax of the transcription, and made decisions to keep the full words rather than the abbreviations, and to remove excerpts that are unknown or uncertain, in an effort to normalize the texts to a common format, as expected in other corpora.

Post Scriptum is a historical Portuguese corpus[5] developed by the Centro de Linguística da Universidade de Lisboa (CLUL) research group. The corpus consists of informal letters in Portuguese and Spanish, most of which are unpublished, written between the 16th and early 20th centuries by authors from different social backgrounds. Due to the nature of the letters, the textual contents are comparable to a spoken corpus, featuring issues from the everyday lives of people from past centuries. We downloaded the whole corpus with POS annotations, for Portuguese and Spanish. The Post Scriptum tagset is a position-based tagset, inspired by the EAGLES tagset for Spanish. The first letter represents the main POS tag, and the second letter represents a fine-grained POS tag. Subsequent letters (or numbers) encode evermore specific information (e.g., person, gender and number). For our purposes, we can convert the tags to the UD tagset by using only the first two letters of the tag. The handling of contractions required minimal effort on our part, since originally these were already kept as single tokens with compound POS tags. Importantly, the corpus features no sentence boundaries, other than the implicit ones between letters, and we used spaCy[6], a popular NLP library that features text segmentation methods, to create the required sentence boundaries.

WikiNER (Nothman et al., 2012) is a silver-standard corpus in multiple languages created from the link structure of Wikipedia. To compensate a lack of annotated data in several languages, the authors proposed to use Wikipedia for automatically generating NER annotations for person, location, organization, and miscellaneous entities. The authors achieve this by classifying Wikipedia articles into named entities, and then converting the links between articles into such entity references, through a set of heuristics specific to Wikipedia. The result was a large scale annotated corpus in nine languages, and for our purposes we downloaded the Portuguese and Spanish sections of the corpus. We parsed the line-based WikiNER format, removing the MISC entities and converting from the IOB scheme to the BIO

[5] http://ps.clul.ul.pt
[6] http://spacy.io/

scheme. With the exception of Colonia, this is the largest corpus used in our experiments, with a total of 3.5M tokens both for Portuguese and Spanish.

Paramopama (Mendonça et al., 2015) is a manually revised corpus derived from the Portuguese WikiNER corpus. The authors revised incorrectly assigned tags in an effort to improve upon the silver-standard WikiNER corpus. Their methodology to create the corpus is as follows: the authors train a NER classifier on the HAREM corpus (Freitas et al., 2010), and use it to automatically label the WikiNER corpus. They then manually analyze the two sets of labels for each sentence and remove the wrong entities. Afterwards, the authors train another NER classifier from the reviewed WikiNER corpus, and use it to label previously unseen sentences extracted from news websites. Similarly to the WikiNER corpus, these sentences are manually reviewed, and the corrected sentences from the Wikipedia and newswire domains are joined to create the Paramopama corpus. The authors also created a version of the HAREM corpus in the CoNLL format, and to avoid repeating past work by preprocessing the HAREM corpus ourselves, we use the Paramopama + second HAREM corpus that the authors produced. We removed time entities from the corpus and converted the raw entities (i.e., no tagging scheme) to the BIO scheme.

The CoNLL-02 shared task (Tjong Kim Sang, 2002) introduced two corpora with named entities in Spanish and Dutch. Together with the English and German corpora from the CoNLL-03 shared task (Tjong Kim Sang and De Meulder, 2003), these corpora are widely used benchmarks in the research community, which serve as a testbed for new model architectures. The Spanish corpus consists of newswire articles from May 2000 made available by the Spanish EFE news agency. In our work, we use the Spanish data with POS tags provided by Xavier Carreras. The POS tags follow the position-based EAGLES tagset for Spanish, and because of this we mostly adapted the conversion tables that we already created for the Post Scriptum corpus. Like in the remaining corpora, we removed all the MISC entities.

AnCora (Taulé et al., 2008) is a multilingual annotated corpus that comprises half a million words in Spanish and in Catalan that were annotated at the morphological, syntactic and semantic levels, thus containing POS and NER annotations. The Spanish corpus consists mainly of newswire texts from the EFE Spanish news agency (225K words) and from the Spanish *El periódico* newspaper (200K words), also including a smaller portion from the *Lexesp* Spanish balanced corpus (75K words). We initially considered using the AnCora-ES version from the UD project which had already been converted to UD guidelines. However, the AnCora UD corpus lacks the named entity annotations which are available in the original corpus. We thus obtained the data for the AnCora corpus from the CoNLL-09 shared task (Hajič et al., 2009). This corpus contains POS tags with morphological features such as gender and number, but in order to convert to the UD tagset, we only required the main and secondary POS tags. AnCora considers six types of named entities: person, organization, location, number, date and other.

Nearly half of the named entities in AnCora are nested and, in our work, we only considered the PER, LOC and ORG entities at the outermost level (e.g., *Electricité de France* is an organization name with a nested location, although we only considered the organization name). MWEs, which in the original AnCora appear as a single token, in our work were expanded into multiple tokens.

The Relaciones Geográficas historical corpus[7] was created in the context of a digital humanities project focusing on the analysis of a collection of 16th century texts known as the *Relaciones Geográficas of New Spain*. The documents describe information with regard to 16th century ethnic groups in Mesoamerica, consisting of a set of answers to questions regarding politics, the natural environment, population history, settlement patterns, native history and customs, etc. A small subset of the data, consisting of approximately 900 sentences, was manually annotated. Though the corpus considers a wide range of named entity categories, which may be potentially nested, we retain only the outermost person, organization and location names.

[7] http://www.lancaster.ac.uk/digging-ecm/corpus/

4

Experimental Evaluation

Contents

This chapter introduces the experimental evaluation of the proposed ideas, first introducing the general evaluation methodology and the evaluation metrics, and then discussing the obtained results.

4.1 Experimental Methodology and Evaluation Metrics

Our overall objective was to assess the usefulness of joint training of cross-lingual neural models for POS tagging and NER, for the Portuguese and Spanish languages, with mixed datasets, domains and tasks. We carried out experiments in four settings: single dataset, cross-domain, multi-task, and cross-lingual. On the single dataset experiments, we evaluate our neural approach independently on each dataset and task. In this context, we also experimented with out-of-domain transfer by using the Portuguese model trained on CINTIL and the Spanish model trained on CoNLL-02, i.e. the largest corpora in Portuguese and Spanish that contain annotations for both tasks, to evaluate the single dataset models on the remaining datasets. In the cross-domain setting, we train models on the aggregation of all datasets for each language and task. In the multi-task setting, we train monolingual models simultaneously targeting POS and NER on all datasets for each language. Finally, in the cross-lingual setting, we train a single model on all Portuguese and Spanish datasets that generalizes across languages, tasks and domains. In all experiments, we chose not to train on the POS annotations of the Colonia corpus. Since the POS tags were automatically generated, and since this corpus is fairly large, it would introduce a large bias on the data quality. We nevertheless chose to evaluate on the Colonia corpus for experiments that use other Portuguese corpora, in order to test the generalization of the models on unseen historical corpora.

To obtain comparable results, we employ the same architecture and hyperparameters among all the settings. The hyperparameter choices were taken from Reimers and Gurevych (2017). In order to model character-level information, we employ a Char-BiLSTM network with a hidden dimension of 25 featuring a character embedding dimension of 30. The word-level BiLSTM network contains two layers, each with a hidden layer dimensionality of 100. We use mini-batches of 32 sentences and employ an early stopping criterion of 10 epochs. We train our model with the Adam optimizer (Kingma and Ba, 2014), a popular variant of stochastic gradient descent which has been broadly adopted for training deep learning models. We use variational dropout (Gal and Ghahramani, 2016) in the LSTM networks and fix the dropout rate to 0.5 both for the linear and recurrent transformations. At the output level, we compute the loss via a single CRF layer for the single dataset and cross-domain settings. On the multi-task and cross-lingual settings, the loss is computed as a sum of two CRF layers, which model label correlations for the tasks of POS and NER. For datasets that only include annotations for one of these tasks, the loss on the other task is masked to prevent it having any impact on network training. In the experiments involving model training with multiple datasets, we also used instance weighting by assigning each sentence to a weight corresponding to one minus the proportion of tokens from the dataset of the sentence, over the total

Language	Dataset	Accuracy			Prec	Rec	F1
		Overall	Unseen	Ambiguous			
Portuguese	Bosque	96.49	91.31	96.32	96.53	96.49	96.45
	CINTIL	98.42	95.96	97.42	98.43	98.42	98.42
	CIPM	97.03	84.74	96.90	97.01	97.03	97.01
	GSD	98.06	95.75	97.57	98.06	98.06	98.06
	Mac-Morpho	97.82	94.95	97.29	97.83	97.82	97.82
	Post Scriptum	92.69	81.85	92.91	92.65	92.68	92.65
	Tycho Brahe	97.46	89.80	97.27	97.46	97.46	97.46
	Average	96.85	90.62	96.53	96.85	96.85	96.84
Spanish	AnCora	98.88	96.59	97.76	98.87	98.88	98.88
	CoNLL-2002	97.84	87.95	96.11	97.83	97.83	97.83
	Post Scriptum	96.97	87.28	97.23	96.95	96.97	96.96
	Average	97.90	90.61	97.03	97.88	97.89	97.89

Table 4.1: POS results obtained from training a model separately for each dataset, together with per-language averages.

number of tokens. Thus, we can counterbalance the impact of larger datasets, which are seen more often during network training, on the remaining corpora, by assigning smaller sentence weights to the larger datasets.

For evaluation we employ seqeval[1], a Python framework for evaluating sequence labeling based on conlleval. For POS tagging, we report different types of accuracy metrics: overall accuracy on all tokens of the test set, accuracy on tokens of the test set not seen during training, and accuracy on ambiguous tokens of the test set, that may have multiple tags on the training set. We also report the micro-averages of precision, recall and F1 scores. For NER, we report the micro-averages of precision, recall and F1 scores at the entity-level (i.e. span-level) and at the token-level. We also report the entity-level and token-level overall accuracies and, for some configurations, we report on per-class results.

4.2 The Obtained Results

Tables 4.1 and 4.2 report the evaluation of the proposed model described in Section 3 on the single dataset setting for POS tagging and NER. We achieve consistent results on the majority of the datasets employing the same general sequence labeling architecture, without tuning hyperparameters on a per-dataset basis. Tables 4.3 and 4.4 present the results of out-of-domain transfer from evaluating models trained on one domain on other domains, and Tables 4.5 and 4.6 similarly present the evaluation of the cross-domain setting for POS tagging and NER. These cross-domain results are generally lower

[1] http://github.com/chakki-works/seqeval

Language	Dataset	Entity Spans				Tokens			
		Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Portuguese	CINTIL	98.23	83.60	85.89	84.73	98.61	98.73	98.70	98.71
	Paramopama	97.50	85.11	88.32	86.69	98.02	98.18	98.17	98.17
	WikiNER	98.71	92.73	93.79	93.26	98.96	99.14	99.13	99.13
	Average	98.15	87.15	89.33	88.23	98.53	98.68	98.67	98.67
Spanish	AnCora	92.32	77.27	80.12	78.67	94.18	94.48	94.37	94.41
	CoNLL-2002	98.04	84.94	88.13	86.51	98.32	98.48	98.42	98.44
	Relaciones Geográficas	98.06	82.29	78.74	80.48	98.64	98.87	98.83	98.83
	WikiNER	98.41	90.71	91.32	91.01	98.72	98.97	98.96	98.96
	Average	96.71	83.80	84.58	84.17	97.46	97.70	97.65	97.66

Table 4.2: NER results obtained from training a model separately for each dataset, together with per-language averages.

than those from the single dataset setting when examining on a per-dataset basis, however they are consistently better than the evaluation on out-of-domain datasets. We argue that although these models perform worse on average, they are able to generalize better, and thus can be more readily applied on unseen texts on various domains than their single dataset counterparts, which fine-tune on one specific domain. Tables 4.7 and 4.8 report the evaluation of the multi-task setting for POS tagging and NER. The results are again lower on average than on the single dataset setting, though not considerably lower when compared to the cross domain models. We maintain the argument of a model being able to generalize across several texts for the multi-task setting. Importantly, due to joint training for POS tagging and NER, our model is now capable of labeling new texts with POS and NER tags on a wide range of domains. Tables 4.9 and 4.10 present the evaluation of the cross-lingual setting, where a single trained model achieves performance on par with or better than the multi-task model for Portuguese datasets. The model performance is worse for Spanish, likely due to the fact that the Spanish embedding space was aligned to the Portuguese embedding space, thus favoring more the Portuguese datasets.

Language	Dataset	Accuracy			Prec	Rec	F1
		Overall	Unseen	Ambiguous			
Portuguese	Bosque	90.30	87.47	88.14	91.55	90.30	88.98
	CINTIL	98.42	95.96	97.42	98.43	98.42	98.42
	CIPM	64.44	54.49	71.73	72.91	64.44	66.48
	Colonia	85.18	53.80	72.17	89.61	85.17	86.78
	GSD	91.96	91.03	89.31	92.78	91.96	91.25
	Mac-Morpho	88.83	86.52	86.57	91.04	88.82	88.30
	Post Scriptum	73.53	55.80	74.75	76.02	73.52	72.15
	Tycho Brahe	86.28	76.12	86.21	86.90	86.27	85.48
	Average	84.87	75.15	83.29	87.40	84.86	84.73
Spanish	AnCora	87.18	64.73	87.70	87.41	87.18	83.22
	CoNLL-2002	97.84	87.95	96.11	97.83	97.83	97.83
	Post Scriptum	74.26	64.25	77.00	75.28	74.26	72.94
	Average	86.43	72.31	86.94	86.84	86.42	84.66

Table 4.3: POS results with a Portuguese model trained on CINTIL and a Spanish model trained on CoNLL-02, for out-of-domain transfer.

Language	Dataset	Entity Spans				Tokens			
		Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Portuguese	CINTIL	98.23	83.60	85.89	84.73	98.61	98.73	98.70	98.71
	Paramopama	95.30	69.66	73.55	71.55	96.14	96.80	96.46	96.56
	WikiNER	95.59	73.06	75.23	74.13	96.34	97.25	96.63	96.78
	Average	96.37	75.44	78.22	76.80	97.03	97.59	97.26	97.35
Spanish	AnCora	76.06	32.56	41.91	36.65	84.67	84.82	86.30	83.68
	CoNLL-2002	98.04	84.94	88.13	86.51	98.32	98.48	98.42	98.44
	Relaciones Geográficas	94.91	39.18	37.87	38.51	96.26	97.45	96.52	96.67
	WikiNER	95.00	67.76	71.26	69.47	95.98	97.08	96.31	96.51
	Average	91.00	56.11	59.79	57.78	93.81	94.46	94.39	93.82

Table 4.4: NER results with a Portuguese model trained on CINTIL and a Spanish model trained on CoNLL-02, for out-of-domain transfer.

Language	Dataset	Accuracy			Prec	Rec	F1
		Overall	Unseen	Ambiguous			
Portuguese	Bosque	92.43	91.08	90.39	92.92	92.43	91.98
	CINTIL	96.05	92.89	94.63	96.33	96.04	96.16
	CIPM	93.53	76.78	93.59	93.73	93.53	93.61
	Colonia	84.90	55.42	71.40	89.81	84.90	86.75
	GSD	92.61	92.74	89.19	94.07	92.61	92.68
	Mac-Morpho	95.27	93.15	93.91	95.66	95.27	95.32
	Post Scriptum	91.39	79.99	91.67	91.37	91.39	91.35
	Tycho Brahe	96.44	89.10	96.19	96.43	96.44	96.42
	Average	92.83	83.89	90.12	93.79	92.83	93.03
Spanish	AnCora	94.43	86.90	93.59	94.72	94.43	94.08
	CoNLL-2002	95.07	82.26	92.03	96.64	95.06	95.71
	Post Scriptum	96.63	85.88	96.97	96.62	96.63	96.62
	Average	95.38	85.01	94.20	95.99	95.37	95.47

Table 4.5: POS results with a cross-domain model trained with all datasets, for each language.

Language	Dataset	Entity Spans				Tokens			
		Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Portuguese	CINTIL	98.28	84.03	85.53	84.77	98.62	98.75	98.73	98.74
	Paramopama	97.37	85.59	87.09	86.34	97.95	98.09	98.10	98.08
	WikiNER	98.43	91.74	92.26	92.00	98.78	98.95	98.95	98.95
	Average	98.03	87.12	88.29	87.70	98.45	98.60	98.59	98.59
Spanish	AnCora	97.93	81.46	87.38	84.32	98.26	98.44	98.34	98.38
	CoNLL-2002	97.77	85.81	86.02	85.91	98.30	98.40	98.41	98.40
	Relaciones Geográficas	98.09	79.87	79.07	79.47	98.59	98.82	98.80	98.81
	WikiNER	97.94	88.22	88.87	88.55	98.35	98.64	98.63	98.64
	Average	97.93	83.84	85.34	84.56	98.38	98.58	98.54	98.56

Table 4.6: NER results with a cross-domain model trained with all datasets, for each language.

Language	Dataset	Accuracy			Prec	Rec	F1
		Overall	Unseen	Ambiguous			
Portuguese	Bosque	91.48	90.29	88.77	92.28	91.48	90.66
	CINTIL	95.07	91.22	93.48	95.57	95.06	95.25
	CIPM	92.09	75.49	92.33	92.39	92.09	92.21
	Colonia	84.82	55.73	71.12	89.83	84.81	86.68
	GSD	92.17	92.06	88.92	93.68	92.17	91.94
	Mac-Morpho	95.21	93.53	93.57	95.54	95.21	95.17
	Post Scriptum	90.89	79.24	91.24	90.85	90.89	90.83
	Tycho Brahe	96.32	88.69	96.09	96.30	96.32	96.29
	Average	92.26	83.28	89.44	93.30	92.25	92.38
Spanish	AnCora	93.25	83.27	93.02	93.78	93.25	92.64
	CoNLL-2002	94.91	81.95	91.73	96.20	94.89	95.45
	Post Scriptum	96.48	86.39	96.74	96.47	96.48	96.47
	Average	94.88	83.87	93.83	95.48	94.87	94.85

Table 4.7: POS results with a multi-task model addressing POS and NER, for each language.

Language	Dataset	Entity Spans				Tokens			
		Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Portuguese	CINTIL	97.52	77.54	80.17	78.83	98.06	98.23	98.18	98.20
	Paramopama	96.97	82.14	84.94	83.52	97.56	97.73	97.74	97.72
	WikiNER	97.99	88.65	90.17	89.40	98.39	98.61	98.60	98.61
	Average	97.49	82.78	85.09	83.92	98.00	98.19	98.17	98.18
Spanish	AnCora	97.73	78.57	85.13	81.72	98.02	98.27	98.12	98.17
	CoNLL-2002	97.64	82.63	84.96	83.78	97.96	98.14	98.09	98.11
	Relaciones Geográficas	97.27	73.82	67.44	70.49	98.04	98.37	98.36	98.31
	WikiNER	97.73	86.59	88.22	87.40	98.19	98.49	98.47	98.48
	Average	97.59	80.40	81.44	80.85	98.05	98.32	98.26	98.27

Table 4.8: NER results with a multi-task model addressing POS and NER, for each language.

Language	Dataset	Accuracy			Prec	Rec	F1
		Overall	Unseen	Ambiguous			
Portuguese	Bosque	91.22	89.16	88.85	91.78	91.22	90.38
	CINTIL	95.38	92.71	93.68	95.83	95.37	95.55
	CIPM	90.10	72.25	91.00	90.37	90.10	90.19
	Colonia	84.91	55.40	71.50	89.95	84.91	86.81
	GSD	92.05	91.54	88.85	93.55	92.05	91.94
	Mac-Morpho	94.84	92.54	93.48	95.27	94.83	94.88
	Post Scriptum	90.92	79.35	91.27	90.89	90.92	90.88
	Tycho Brahe	96.36	88.48	96.16	96.35	96.36	96.34
	Average	91.97	82.68	89.35	93.00	91.97	92.12
Spanish	AnCora	92.76	84.29	91.10	93.26	92.76	91.95
	CoNLL-2002	93.46	78.13	88.61	95.68	93.44	94.39
	Post Scriptum	95.51	85.31	95.85	95.50	95.51	95.50
	Average	93.91	82.58	91.85	94.81	93.90	93.95

Table 4.9: POS results obtained with a single cross-lingual model trained on all datasets, tasks and languages.

Language	Dataset	Entity Spans				Tokens			
		Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Portuguese	CINTIL	97.61	77.72	80.34	79.01	98.12	98.31	98.26	98.28
	Paramopama	97.10	82.70	85.59	84.12	97.71	97.90	97.91	97.90
	WikiNER	98.17	90.13	91.23	90.68	98.56	98.77	98.76	98.77
	Average	97.63	83.52	85.72	84.60	98.13	98.33	98.31	98.32
Spanish	AnCora	78.73	36.79	45.00	40.48	86.36	86.45	87.69	85.93
	CoNLL-2002	94.58	74.01	72.26	73.12	94.92	96.59	95.45	95.83
	Relaciones Geográficas	95.90	59.93	53.16	56.34	97.09	97.39	97.54	97.44
	WikiNER	97.70	86.78	88.04	87.41	98.16	98.49	98.47	98.48
	Average	91.73	64.38	64.62	64.34	94.13	94.73	94.79	94.42

Table 4.10: NER results obtained with a single cross-lingual model trained on all datasets, tasks and languages.

Language	Portuguese								Spanish		
Dataset	Bosque	CINTIL	CIPM	Colonia	GSD	Mac-Morpho	Post Scriptum	Tycho Brahe	AnCora	CoNLL-2002	Post Scriptum
ADJ	84.50	91.02	62.26	68.72	87.41	90.06	77.89	90.86	87.88	83.86	83.48
ADP	96.76	98.30	95.67	94.83	95.87	95.71	95.23	98.21	95.33	99.44	99.15
ADP+ADV	100.00	97.67	57.14	81.16	100.00	93.94	90.91	91.83	-	-	-
ADP+DET	98.09	99.44	93.00	95.17	97.86	97.73	91.78	99.06	37.02	-	98.88
ADP+PRON	58.82	85.71	77.19	46.98	68.85	83.50	91.10	96.44	-	-	93.83
ADV	89.46	94.37	80.55	88.42	90.37	87.05	92.50	95.22	97.20	97.21	95.20
AUX	24.00	55.43	-	-	26.59	-	85.35	91.03	98.63	99.09	96.52
CCONJ	86.45	89.87	95.42	89.08	86.86	91.14	95.70	96.86	99.18	99.29	98.74
DET	96.92	94.27	80.10	87.96	97.01	95.98	89.47	96.50	98.20	99.12	97.06
INTJ	-	81.77	72.73	73.11	-	67.10	74.70	86.24	36.36	0.00	81.08
NOUN	94.73	97.28	88.98	81.51	95.48	96.35	89.30	96.24	91.05	90.85	92.22
NUM	84.29	96.08	84.76	80.44	89.29	89.95	91.50	97.87	79.16	95.63	96.28
PRON	81.31	84.29	87.05	78.09	82.11	78.90	88.80	94.98	93.21	93.43	93.22
PROPN	93.24	95.63	87.55	-	94.26	95.04	83.68	94.68	69.41	23.47	91.58
PUNCT	99.96	99.83	99.95	99.52	99.54	99.95	95.04	99.75	99.99	99.96	99.69
SCONJ	52.97	12.24	-	38.19	-	74.37	86.97	82.89	91.87	89.77	92.05
SYM	57.78	71.51	-	-	62.86	95.79	-	-	-	-	-
VERB	81.29	95.72	90.35	88.55	84.73	96.53	93.81	97.59	98.78	92.35	96.57
VERB+PRON	100.00	99.46	49.12	90.01	98.77	99.78	85.81	97.78	0.00	-	-
X	0.00	-	-	0.00	0.00	-	2.06	31.43	0.00	0.00	0.00

Table 4.11: F1 scores per POS tag category, obtained with a single cross-lingual model.

		Entity Spans			Tokens		
Language	Dataset	LOC	ORG	PER	LOC	ORG	PER
Portuguese	CINTIL	73.67	72.09	86.01	76.03	76.58	87.31
	Paramopama	88.10	69.55	86.65	90.48	72.39	89.05
	WikiNER	91.68	79.51	92.38	94.03	80.25	93.55
	Average	84.48	73.72	88.35	86.85	76.41	89.97
Spanish	AnCora	43.20	31.00	47.98	49.89	51.82	53.11
	CoNLL-2002	76.89	70.76	72.03	77.79	79.67	67.15
	Relaciones Geográficas	58.58	0.00	53.78	66.13	0.00	76.92
	WikiNER	87.29	73.69	91.71	91.17	77.99	93.57
	Average	66.49	43.86	66.38	71.24	52.37	72.69

Table 4.12: F1 scores per named entity type at the span- and token-level, obtained with a single cross-lingual model.

5

Conclusions and Future Work

Contents

This chapter provides a summary of the findings from this dissertation. The chapter then concludes with a discussion of directions for future work.

5.1 Conclusions

In this work we gathered a comprehensive list of datasets in Portuguese and Spanish, comprising historical and modern texts with parts-of-speech and named entity annotations. We standardized these corpora to a common data format, converted the POS tags to the Universal POS tagset and NER tags to the BIO tagging scheme featuring person, organization and location entities. We evaluated the performance of a modern sequence labeling model under diverse transfer learning settings. We experimented with cross-domain, multi-task and cross-lingual transfer, following recently proposed ideas from the state-of-the-art. In particular, our cross-lingual model achieves 91.97% of overall accuracy and 84.60% of entity-level F1 score for Portuguese, and 93.91% of overall accuracy and 64.34% of entity-level F1 score for Spanish, when averaging over all datasets for these languages. Though the results from transfer learning are still far from the state-of-the-art on these datasets, we argue that these models would fare better for processing previously unseen texts. We also hope to bring more attention to these understudied languages and domains by releasing the code that standardizes the datasets.

5.2 Future Work

Despite the interesting results, there are also many possibilities for future work. Besides considering Portuguese and Spanish, we can perhaps also consider experimenting simultaneously with data from multiple similar languages (Rahimi et al., 2019). Specifically, our cross-lingual approach could perhaps benefit from datasets in Galician, which would act as a bridge between Spanish and Portuguese due to the similarities between these languages. Another interesting direction is to jointly train POS tagging and NER simultaneously with other tasks. For example, our sequence tagger can be extended to predict the next word and the previous word in the sentence, in addition to the POS and NER tags of the current word. A language modeling auxiliary objective, as previously explored by Rei (2017), could provide slight improvements in performance without requiring additional training data. We can consider experimenting with recently proposed architectures for sequence labeling, for instance replacing standard RNN cells with alternatives that deepen the state transition path at each position, and/or adding to the representation of each token a global representation learned from the entire sentence (Liu et al., 2019). Another promising line of work is cross-lingual alignment of contextual embeddings such as ELMo or BERT, using these instead of the FastText embeddings. We believe that the recent successes of contextual embeddings in monolingual models can be extended to cross-lingual settings (Schuster et al., 2019), through

more effective representations of language arising from context-awareness in cross-lingual models.

Besides NER and POS tagging, and following similar ideas related to the combination of existing resources within a cross-lingual setting, we also plan to experiment with the training of neural models for co-reference resolution (i.e., for linking together mentions that relate to the same real world entities, including not just the named entity mentions present in the text but also pronouns and other words that are potentially referring to entities). Co-reference resolution is also a rather old NLP research topic that has witnessed a revival of interest in connection to modern deep learning approaches (Clark and Manning, 2016a,b), although with few previous studies focusing on Spanish or Portuguese texts.

Bibliography

S. Aluísio, J. Pelizzoni, A. R. Marchi, L. de Oliveira, R. Manenti, and V. Marquiafável. An account of the challenge of tagging a reference corpus for Brazilian Portuguese. In *Proceedings of the International Conference on Computational Processing of the Portuguese Language*, 2003.

F. Barreto, A. Branco, E. Ferreira, A. Mendes, M. F. Nascimento, F. Nunes, and J. Silva. Open resources and tools for the shallow processing of Portuguese: The TagShare project. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2006.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3:1137–1155, 2003.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 2017.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

J. P. C. Chiu and E. Nichols. Named entity recognition with bidirectional LSTM-CNNs. *CoRR*, abs/1511.08308, 2015.

K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

K. Clark and C. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016a.

K. Clark and C. Manning. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2016b.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning*, 2008.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398, 2011.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith. Transition-based dependency parsing with stack long short-term memory. *CoRR*, abs/1505.08075, 2015.

S. Eger, P. Youssef, and I. Gurevych. Is it time to swish? Comparing deep learning activation functions across NLP tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

E. R. Fonseca, J. L. G. Rosa, and S. M. Aluísio. Evaluating word embeddings and a revised corpus for part-of-speech tagging in Portuguese. *Journal of the Brazilian Computer Society*, 21(1):2, 2015.

C. Freitas, P. Carvalho, H. Gonçalo Oliveira, C. Mota, and D. Santos. Second HAREM: Advancing the state of the art of named entity recognition in Portuguese. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2010.

C. Freitas, L. F. Trugo, F. Chalub, G. Paulino-Passos, and A. Rademaker. Tagsets and datasets: Some experiments based on Portuguese language. In *International Conference on Computational Processing of the Portuguese Language*, 2018.

Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.

Y. Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1), 2017.

A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer, 2012.

K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. *CoRR*, abs/1503.04069, 2015.

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, et al. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Conference on Computational Natural Language Learning*, 2009.

K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. *CoRR*, abs/1611.01587, 2016.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

T. Horsmann and T. Zesch. Do LSTMs really work so well for PoS tagging?–A replication study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017.

Z. Huang, W. Xu, and K. Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.

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

W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso. Finding function in form: Compositional character models for open vocabulary word representation. *CoRR*, abs/1508.02096, 2015.

Y. Liu, F. Meng, J. Zhang, J. Xu, Y. Chen, and J. Zhou. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019.

X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2016.

R. McDonald, J. Nivre, Y. Quirmbach-Brundage, Y. Goldberg, D. Das, K. Ganchev, K. Hall, S. Petrov, H. Zhang, O. Täckström, et al. Universal dependency annotation for multilingual parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2013.

C. Mendonça, Jr, H. Macedo, T. Bispo, F. Santos, N. Silva, and L. Barbosa. Paramopama: A Brazilian-Portuguese corpus for named entity recognition. In *Actas do Encontro Nacional de Inteligência Artificial e Computacional*, 2015.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2016.

J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175, 2012.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, 2013.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

B. Plank, A. Søgaard, and Y. Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *CoRR*, abs/1604.05529, 2016.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

A. Rademaker, F. Chalub, L. Real, C. Freitas, E. Bick, and V. de Paiva. Universal dependencies for Portuguese. In *Proceedings of the International Conference on Dependency Linguistics*, 2017.

A. Rahimi, Y. Li, and T. Cohn. Multilingual NER transfer for low-resource languages. *CoRR*, abs/1902.00193, 2019.

M. Rei. Semi-supervised multitask learning for sequence labeling. *CoRR*, abs/1704.07156, 2017.

N. Reimers and I. Gurevych. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017.

S. Ruder, A. Søgaard, and I. Vulic. A survey of cross-lingual embedding models. *CoRR*, abs/1706.04902, 2017.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

C. D. Santos and B. Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the International Conference on Machine Learning*, 2014.

C. N. d. Santos and V. Guimaraes. Boosting named entity recognition with neural character embeddings. *CoRR*, abs/1505.05008, 2015.

T. Schuster, O. Ram, R. Barzilay, and A. Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *CoRR*, abs/1902.09492, 2019.

A. Søgaard and Y. Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2016.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

M. Taulé, M. A. Martí, and M. Recasens. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2008.

E. F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the Conference on Natural Language Learning*, 2002.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Conference on Natural Language Learning*, 2003.

M. F. Xavier, M. T. Brocardo, and M. d. G. Vicente. CIPM–Um corpus informatizado do português medieval. *Actas do Encontro da Associação Portuguesa de Linguística*, 2:599–612, 1994.

Z. Yang, R. Salakhutdinov, and W. W. Cohen. Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270, 2016.

Z. Yang, R. Salakhutdinov, and W. Cohen. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*, 2017.

M. Zampieri and M. Becker. Colonia: Corpus of historical Portuguese. *ZSM Studien, Special Volume on Non-Standard Data Sources in Corpus-Based Research*, 5, 2013.

C. Zhou, X. Ma, D. Wang, and G. Neubig. Density matching for bilingual word embedding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.