Part-of-Speech Embeddings for Portuguese

Paulo L. Medeiros, Bledson Bezerra, Carlos Prolo, Antônio Thomé

[1]Departamento de Informática e Matemática aplicada
Universidade Federal do Rio Grande do Norte (UFRN) – Natal– RN – Brazil

{pauloaugusto99, bledson}@ufrn.edu.br, {prolo, thome}@dimap.ufrn.br

Abstract. *Training classification models on multiple datasets is a common procedure with Deep Learning, since better results are expected with more data. The usual approach is to merge all datasets, mapping each of the original class sets to a single one. However, for Part-of-Speech tagging, where dataset annotation is a quite theoretically-subjective task, there is not always an explicit correspondence between two different tag sets. Moreover, it strongly depends on the tokenization assumptions made in each corpus. Along with some of the most recent deep learning techniques (Bi-LSTMs stacking, word and char embeddings, residual connections), we introduce an approach to POS tagging learning that conforms to multiple tagsets with different tokenization assumptions from different training corpora. Crucial to the approach is the introduction of the concept of intermediate continuous distributed POS representations. Without the need of pretraining, we achieve state-of-the-art accuracy while building a robust versatile POS tagger. We suggest that, for downstream applications, the POS embeddings can be used instead of the POS tags.*

1. Introduction

Word embeddings have been successfully used to represent words and word senses in Natural Language Processing (NLP) applications. They became popular after advances in neural computing made them feasible to be computed [Bengio et al. 2003, Collobert et al. 2011, Mikolov et al. 2013a, Mikolov et al. 2013b, Pennington et al. 2014, Jurafsky 2000]. The notion of distributed representation made it possible to insert a notion of relatedness among words, which is not implicit in their morphology and is difficult to be captured using manual extracted features. Once this type of representation started being computationally feasible and popularized, all tasks in NLP started being revisited, gaining accuracy by including embeddings in preexisting algorithms.

There are several other concepts in linguistics, involving different kinds of taxonomies, that may as well be able to benefit from the idea of distributed representations. In this work we focus in part-of-speech tagging. It is intuitive and appealing the notion that words in a certain context can be classified into classes that represent their syntactic distributional properties. However it is hard to define a specific and finite tagset that solves all theoretical and practical concerns that have been raised over all these years. So, several corpora have been built with rather distinct tagsets which are hard to map into one another. [Aluísio et al. 2003, Afonso et al. 2002, Petrov et al. 2012, Marcus et al. 1993, Francis and Kucera 1979] Moreover each of these tagsets are built based on specific assumptions of tokenization as, for instance, whether contractions shall be split or not.

Based on well known models for POS tagging, such as [Plank et al. 2016], [Ling et al. 2015], using a bidirectional long short-term memory (Bi-LSTM) architecture [Graves and Schmidhuber 2005, Hochreiter and Schmidhuber 1997], and recent concepts in neural computing, such as word and character embeddings, Bi-LSTM stacking and residual connections [Lample et al. 2016, Dos Santos and Zadrozny 2014, Mikolov et al. 2013a, Mikolov et al. 2013b, Mikolov et al. 2018, Peters et al. 2018, He et al. 2015], we built a neural architecture for POS tagging centered in the idea of continuous distributed representation for POS. The code and all related material necessary to reproduce the experiments reported here are available at `https://github.com/pauloamed/STIL2019`.

2. The architecture

2.1. Bi-LSTM architecture

Bi-LSTM is one of the most used deep learning architectures on NLP nowadays. It is a bidirectional extension [Graves and Schmidhuber 2005] of the LSTM [Hochreiter and Schmidhuber 1997] that itself is an extension of the Elman Recurrent Neural Network (RNN) [Elman 1990].

2.1.1. Elman Recurrent Neural Networks

RNNs act as a *feature extractor* of a non-fixed-size sequence [Elman 1990, Goldberg 2017, Graves 2012], by mapping the entire *history* of previous time-steps to the current one. This mapping is done by keeping a *memory* vector running and being updated through the time-steps. However, one should note that the parameters used to update this memory are the same regardless of the time-step, giving generalization power to the architecture through the sequence. When dealing with natural language sentences, the time steps should be regarded, instead, as positions in the input sentences, either words or characters depending on the desired granularity.

Let θ be the model's parameters, n the sequence size, and x_i, h_i and y_i the input, the memory and the output vectors, respectively, at position i. Also let f and g be non-linear mappings using θ. We can represent an RNN model as follows

$$\begin{aligned} &h_0 = \mathbf{0}, \text{ the null vector} \\ &h_i = f(x_i, h_{i-1}; \theta), 1 \leq i \leq n \\ &y_i = g(h_i; \theta), 1 \leq i \leq n \end{aligned}$$

2.1.2. Long short-term memory and its bidirectional extension

LSTM was designed to address the problem of vanishing/exploding gradients in RNNs. It is a more complex architecture, where units in the hidden layer are replaced by subnets, called memory blocks. These blocks enjoy of multiplicative gates, which, along with an efficient training algorithm, allow for a long term memory and prevent the vanishing/exploding gradient problem.

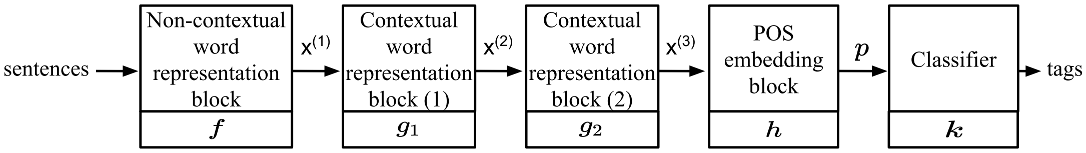

sentences
Non-contextual word representation block
f
$x^{(1)}$
Contextual word representation block (1)
g_1
$x^{(2)}$
Contextual word representation block (2)
g_2
$x^{(3)}$
POS embedding block
h
p
Classifier
k
tags

Figure 1. Architecture blocks

Bidirectional recurrent architectures [Graves and Schmidhuber 2005] were designed for tasks where the output of the current time-step also depends on future time-steps. An additional network with same architecture is instantiated which processes time-steps on the reverse order: from t_n to t_1. Thus, the outputs of both networks are combined into one vector, and this will be the output of the bidirectional network.

2.2. Residual connections

Sometimes, the optimal mapping that a layer (or consecutive layers) need to compute is the identity, or a similar mapping. Based on this, residual connections, firstly designed for CNNs (Convolutional Neural Networks) [He et al. 2015], follow the hypothesis that a zero mapping is easier to approximate than an identity mapping and represents a "pass" on the model architecture from a lower to a higher layer. By doing this, it expects the affected layers to compute a null vector.

2.3. Our architecture

We organized our architecture in blocks (Figure 1). The initial blocks were designed as a feature extractor for the words in context. Block f computes a non-contextual char-embedding-based word representation for every word in the sentence, while g_1 and g_2 compute contextual representations for them, that is, word senses. The next two blocks (h, k) are used to compute POS-refined representations for every word sense, classifying them according to the dataset from which the sentence was extracted. This is explained in detail below. Blocks f, g_1 and g_2 could be pretrained using a language modeling task.

Character embeddings have size d_c, word and word sense representations computed by f, g_1 and g_2 have size d_w, and the ones computed by h have size d_t. POS embeddings are represented as p. Sequences such as $x_1, x_2, ..., x_m$ will be referred to as $x_{1:m}$. We refer to n as the sentence size.

2.3.1. Non-contextual representations of words

Many approaches have been developed to compute non-contextual representations of words, i.e., representations providing specifically intraword information, fundamental when dealing with morphologically rich languages, such as Portuguese. [Dos Santos and Zadrozny 2014] uses a convolutional network over character embeddings to obtain word representations. This idea is also used on [Peters et al. 2018] in a more complex manner. It has also been proposed to encode character embeddings using Bi-LSTMs [Lample et al. 2016, Ling et al. 2015], and also adding all the character n-gram vectors in each word. [Mikolov et al. 2018]

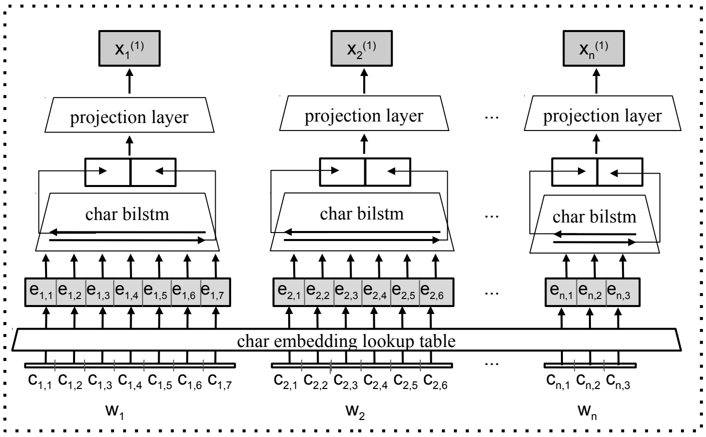

x1(1)
x2(1)
xn(1)
projection layer
projection layer
projection layer
...
char bilstm
char bilstm
char bilstm
e1,1 e1,2 e1,3 e1,4 e1,5 e1,6 e1,7
e2,1 e2,2 e2,3 e2,4 e2,5 e2,6
en,1 en,2 en,3
char embedding lookup table
c1,1 c1,2 c1,3 c1,4 c1,5 c1,6 c1,7
c2,1 c2,2 c2,3 c2,4 c2,5 c2,6
cn,1 cn,2 cn,3
w1
w2
wn

Figure 2. f**: Character-based non-contextual word representation block**

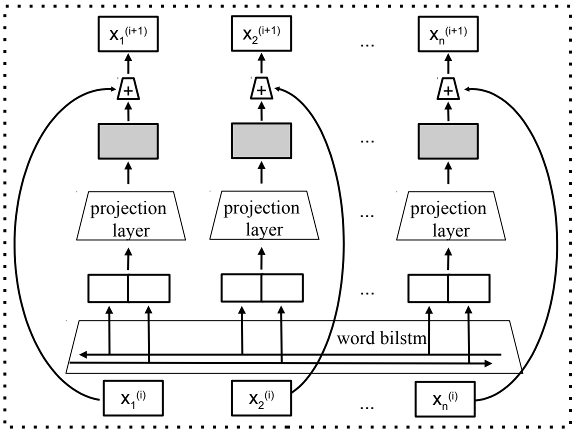


Figure 3. $g_i, i \in \{1, 2\}$**: Contextual word representation block**

As shown in Figure 2, each word is mapped into a character embedding sequence. Consider w_i the $i^{th}, 1 \leq i \leq n$, word of the sentence and $c_{i,j}$ its j^{th} character. For every character $c_{i,j}$, an embedding $e_{i,j} \in \mathbb{R}^{d_c}$ is computed by a lookup table trained along with the whole model. For every word, their character embeddings feed a Bi-LSTM and the last output of the forward and backward passes of this Bi-LSTM are extracted and concatenated, composing an intermediate representation in $\mathbb{R}^{2d_w}$. These representations will then feed a linear layer (dimensionality reduction) to produce $x^{(1)}_{1:n}$, vectors in $\mathbb{R}^{d_w}$.

2.3.2. Contextual representations of words

Contextual representation of words, also called word senses, take into account not only the words in isolation but also their context. In our model, these representations depend on the input from the entire sentence, unlike fixed-sized windows techniques [Mikolov et al. 2013a, Goldberg 2017]. This gives the models much more power to learn better representations.

Inspired by ELMo [Peters et al. 2018], we stack two instances (g_1 and g_2) of the same block architecture (Figure 3) to compute these representations. Block g_i, $i \in \{1, 2\}$,

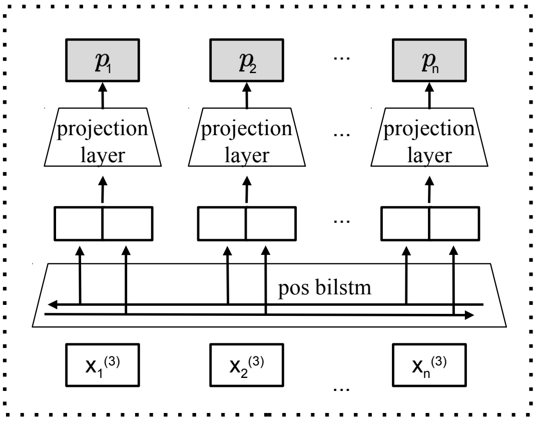
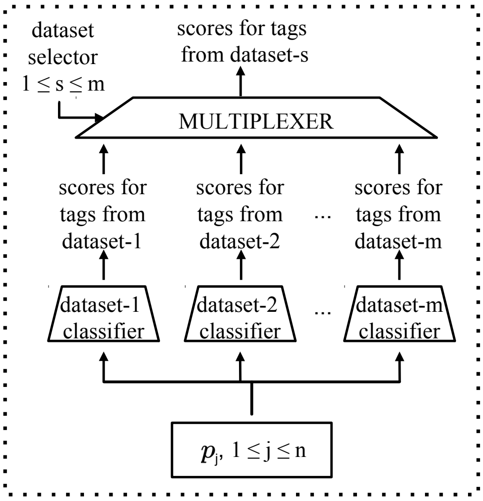


Figure 4. h**: POS-refined word representation block**



dataset selector $1 \leq s \leq m$
scores for tags from dataset-s
MULTIPLEXER
scores for tags from dataset-1
scores for tags from dataset-2
scores for tags from dataset-m
dataset-1 classifier
dataset-2 classifier
dataset-m classifier
p_j, $1 \leq j \leq n$

Figure 5. k**: Classifier**

takes as input $x^{(i)}_{1:n}$, and passes it through a Bi-LSTM layer. For each word position, the forward and backward outputs of the Bi-LSTM are concatenated into an intermediate representation in $\mathbb{R}^{2d_w}$. These will then feed a linear layer in $\mathbb{R}^{d_w}$, computing intermediate representations which are added element-wise to $x^{(i)}_{1:n}$ to compute $x^{(i+1)}_{1:n}$.

2.4. POS-refined representations of words and classification

The final representations $p_{1:n}$ computed by the model are the POS embeddings we have been referring to throughout the paper, as refinements of the word sense embeddings. They are computed by block h (Figure 4), almost the same architecture as g, but without the residual connections. The word sense representations $x^{(3)}_{1:n}$ will feed the Bi-LSTM. For each word, the forward and backward outputs are concatenated, composing embeddings in $\mathbb{R}^{2d_t}$. A linear projection transforms them into $p_{1:n}$ in $\mathbb{R}^{d_t}$.

As $x^{(3)}_{1:n}$, $p_{1:n}$ are still continuous distributed representations of the word in context. However they are peculiar, as it is expected that their training stresses the syntactic information of the words in context, since a single projection layer should be able to map them into conventional POS tags. This is highly unexpected unless they already represent the POS but in a distributed continuous manner in the sense of [Harris 1954] and [Firth 1957]. Note also that these cannot be thought as ”POS tag” embeddings, since they are not computed as if by a look up table with a finite tagset as input. And, on top of that, they strongly take context into account.

Figure 5 shows how this is accomplished. The $p_{1:n}$ embeddings are fed into projection layers followed by softmax normalisation, one for each dataset. Then, during training, we select the output corresponding to the tagset used in the corpora from which the sentence came from, and compare it to the target, computing the error used by the optimisation process. The error measure used is the cross entropy.

When used later in downstream applications, the POS-refined embeddings could be directly feed into the upper modules, instead of using the more limited POS tags.

Dataset name	Tagset	Train set		Validation set		Test set	
		#sent	#words	#sent	#words	#sent	#words
Mac-morpho	MM	37948	728497	1997	38881	9987	178373
GSD-UD	UD	9664	255755	1210	32129	1204	31496
Bosque-UD	UD	8328	206744	560	10851	477	10199
Bosque-LT	LT	3355	65086	419	7229	420	7995
Total	All	59295	1256082	4186	89090	12088	228063

Table 1. The datasets

3. Evaluation

The architecture used in this paper was implemented using *pytorch* [Paszke et al. 2017]. In this section, we depict the experiments setups, datasets details and results.

3.1. Datasets

The datasets we used are listed in Table 1.

Mac-Morpho: This is a large corpus of Brazilian Portuguese from newspaper articles annotated with POS tags with its own tagset, referred to in Table 1 as *MM* [Aluísio et al. 2003, Fonseca and Rosa 2013, Fonseca et al. 2015].[1] We used its third revision with 26 POS tags (22 primitive tags, plus punctuation, plus three tags for contractions such as *PREP+ART* for the word *”das”*, which is a contraction of the preposition *”de”* [”of” in English] and the article *”as”* [”the”]). Training, validation and test sets were taken from the three files provided for this purpose in the site.

GSD-UD: This is a corpus of Brazilian Portuguese available in `https://universaldependencies.org/#download` converted from the Google Universal Dependencies Treebank, annotated with the 16 tags plus punctuation extended POS tagset of [Petrov et al. 2012] (UPOS). We used the training/dev/test files from version 2.4.

Bosque-UD: Bosque is part of *Floresta Sintática* a well known collection of corpora in the *Linguateca*[2]. It contains annotated corpora of Brazilian and European Portuguese [Afonso et al. 2002]. Bosque-UD is a later version converted to UD POS tags [Rademaker et al. 2017]. We used version 2.3, available at `https://universaldependencies.org/#download`.

Bosque-LT: This is an older version of Bosque with the original tagsets in [Afonso et al. 2002]. We used the Brazilian Portuguese corpus of newspaper articles from *A Folha de São Paulo*. It has its own tagset with 23 tags plus punctuation. The sentences are not the same as those in Bosque-UD but there was a substantial overlap. For the sake of fairness, we built the training, development and test set by first moving sentences which were also in Bosque-UD to the same corresponding training/development/test sets. The remaining sentences in Bosque-LT were distributed to fit a 80-10-10 % split. [3]

[1] See also `http://nilc.icmc.usp.br/macmorpho/` and `http://nilc.icmc.usp.br/macmorpho/macmorpho-manual.pdf`.

[2] See `https://www.linguateca.pt/Floresta/corpus.html#bosque`

[3] We noticed a small overlap remained between Mac-morpho and Bosque that crosses from training to test or development files, involving no more than 10 noisy sentences. Their impact in evaluation is negligible.

Dataset	Single corpus		Combined Training	
	μ	σ	μ	σ
Mac-Morpho	97.28	0.039	**97.46**	0.004
GSD-UD	97.27	0.024	97.87	0.034
Bosque-UD	96.24	0.097	97.18	0.039
Bosque-LT	96.40	0.102	98.32	0.226
Micro-average	97.20	-	97.53	-

Table 2. Accuracies

Due to the different tokenization assumptions in each corpora, our model learns representations from all word forms involved. Recalling the previous example, it will successfully encode both the contracted word *"das"* and their split pair *"de"* and *"as"*. That allows for the versatile use of the POS tagging module embedded in downstream applications with any tagset assumption.

It is interesting to point out that whereas the correspondence from one tagset to another is not trivial when translating, say, from LT to UD, and even the choice of the correct POS tag by annotators is a challenging problem reported on any corpus annotator manual, the continuous distributed intermediate representation for the POS is allowed to freely represent the syntactic distribution of the word in context.

3.2. Experiments and results

Each experiment consisted of a training phase of 55 epochs and a test phase. We validated the parameters at the end of each epoch by calculating the average error over some development set. At the end of training we chose the parameter set giving the smallest error. Each experiment was performed three times and we report the average accuracy and standard deviation in Table **??**. Column labelled "Single Corpus" reports the first set of four experiments, one for each of the original corpora, used for control. Each cell in this column reports the accuracy obtained when the model was trained, validated and tested with sentences of a specific corpus.

In the other experiment reported under "Combined Training", we trained the model using the sentences of all of the four corpora, informing, at the multiplexer, the corpus (and hence the tagset) the sentence came from. Validation was done over the union of the development sets as well. Each cell in that column reports the accuracy of the combined model when tested over the test set for each specific corpus. The bottom row has the micro-average accuracy.

For the sake of reproducibility, batches of 32 sentences were defined at the beginning of each experiment, each batch contained sentences from the same corpus (even in the combined training). We shuffled the order of the batches for each epoch. We used Adadelta optimizer [Zeiler 2012], with learning rate 1.0, $\rho = 0.9$, and $\epsilon = 1\mathrm{e}{-6}$. For all experiments, $d_c = 70$, $d_w = 350$ and $d_t = 150$. Dropout [Srivastava et al. 2014] was used, and the layers in which it was applied are the ones pointed out with a gray filling on Figures 2, 3, 4 and 5. Drop ration was set to 0.1 in f, 0.2 in g_1 and g_2, and 0.4 in h. We used begin/end delimiters for words (BOW, EOW) and sentences (BOS, EOS). We did not pretrain the word representation level.

Crucially, the results obtained when training on all corpora, mixing different tagsets and different tokenization assumptions, not only gives higher micro-average accuracy, but surpasses the accuracy on all test sets.

But more importantly we obtain an instance of representation of the parts of speach, the POS embeddings p_i, which captures the syntactic notion of lexical category of the words in context, which are not bound to the specific finite tagsets and are more likely to represent a true word class distribution.

4. Related Word

[Bick 2000] is a handwritten rule-based parser that for long time was probably the best choice for reasonably accurate POS tagging for Portuguese, with the tagset from the VISL Project used in the Linguateca (LT in Table 1). With the advent of the neural approaches revisiting POS tagging, [Dos Santos and Zadrozny 2014] achieved accuracy of 97.47% on the Mac-Morpho corpus revision 1, with a character-based convolutional neural network (CNN) approach, while [Fonseca et al. 2015] reached 97.33% in Mac-Morpho's revision 3, which was the one used in our experiments.[4] [Plank et al. 2016] multilingual architectures reports 97.94% on Bosque-UD v1.2 (available in `https://universaldependencies.org/#download`).

Interestingly enough, our approach generates embeddings for word forms under each tokenization assumed in each of the corpus used. Then maps these word forms in context to continuous universal distributed tags that accurately adapt to all of the tagsets of these corpora; and it still provides an intermediate instance that is free to represent the true distribution of lexical categories of the words in context, which can be used in downstream applications that benefits from POS tagging.

5. Conclusions

In this paper we presented a neural architecture that generates intermediate continuous distributed representations for Portuguese POS which accurately adapt to different tagsets with a single neural output layer, still being allowed to preserve the true distribution of the word classes in context in the sense of [Harris 1954] and [Firth 1957].

The underlying embeddings generated as byproduct represent word forms under any of the tokenization assumptions assumed by each of the corpora. Likewise the intermediate tag representations, therefore, are able to support the classification of any of these forms, including multiword expressions as named entities, composing a very robust instance for plugging into downstream applications.

We think of these intermediate tag embeddings as real universal POS tag representations and plan to extended the concept soon in the future including corpora for multiple languages and testing extrinsically on downstream application.

References

Afonso, S., Bick, E., Haber, R., and Santos, D. (2002). Floresta sintá(c)tica: A treebank for portuguese. In *Proceedings of the Third International Conference on Language Re-*

[4] Revision 1 used in [Dos Santos and Zadrozny 2014] was slightly easier because contractions were split up. In that revision [Fonseca et al. 2015] obtained 97.57% accuracy.

sources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain.

Aluísio, S., Pelizzoni, J., Marchi, A. R., de Oliveira, L., Manenti, R., and Marquiafável, V. (2003). An account of the challenge of tagging a reference corpus for brazilian portuguese. In Mamede, N. J., Trancoso, I., Baptista, J., and das Graças Volpe Nunes, M., editors, *Computational Processing of the Portuguese Language*, pages 110–117, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bick, E. (2000). *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. PhD thesis, Århus.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Dos Santos, C. and Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In *31st International Conference on Machine Learning, ICML 2014*, volume 5.

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

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32.

Fonseca, E. R. and Rosa, J. L. G. (2013). Mac-morpho revisited: Towards robust part-of-speech tagging. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology, STIL 2013, Fortaleza, Brazil, October 21-23, 2013*.

Fonseca, E. R., Rosa, J. L. G., and Aluísio, S. M. (2015). Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. *J. Braz. Comp. Soc.*, 21(1):2:1–2:14.

Francis, W. N. and Kucera, H. (1979). Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Goldberg, Y. (2017). *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 1th edition.

Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pages 5–6.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Ling, W., Dyer, C., Black, A. W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L., and Luis, T. (2015). Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

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

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *LREC*. European Language Resources Association (ELRA).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL.

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

Petrov, S., Das, D., and McDonald, R. T. (2012). A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2089–2096.

Plank, B., Søgaard, A., and Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *ACL (2)*. The Association for Computer Linguistics.

Rademaker, A., Chalub, F., Real, L., Freitas, C., Bick, E., and de Paiva, V. (2017). Universal dependencies for portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling)*, pages 197–206, Pisa, Italy.

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

Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.