The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

Semantic Labeling of English Texts with Ontological Categories Employing Recurrent Networks

Roberta Caroline Rodrigues Silva, Alcione de Paiva Oliveira, Alexandra Moreira
Universidade Federal de Viçosa, Departamento de Informática,
36570-900 Viçosa, Brazil

Abstract

Semantic labeling of texts allows people and computing devices to more easily understand the meaning of a natural language sentence as a whole.Semantic annotation is often one of the first steps carried out by applications focused on natural language processing. However, this step is often done manually, which is very expensive and time-consuming. When automatic methods are employed they require that a set of features, elaborated by specialists, be provided so that the system can assign probabilities in order to make inferences. In this article we present a model of the deep recurrent network that semantically annotates texts in English using as labels the top categories of an ontology. The tests showed that it is possible to obtain better results than the models that need the features to be made explicit.

Keywords: Natural language Processing, semantic annotation, recurrent network, LSTM, Ontology

Introduction

The understanding of natural language is only possible from the moment one understands the meaning of lexical elements in the context of a statement. This understanding, which naturally occurs to people, has only recently presented encouraging results for automatic systems. This is due to the difficulty of associating the precise meaning to syntactic elements. The attribution of meaning to lexical items is complex and is directly linked to the relative positioning of one word in relation to the others. Traditionally, the assignment of meaning to lexical items, so that utterances can be processed by machines, is done manually by specialists in a process called semantic annotation or semantic labeling. According (Pustejovsky and Stubbs 2012) any metadata tag used to mark up elements of the dataset is called an annotation over the input. Normally, the annotation process is expensive and time consuming, being done manually by specialists. However, recent advances in hardware and machine learning techniques, especially those of deep learning, have opened up new perspectives for automatic semantic annotation. When automatic labeling systems are employed, methods such as maximum entropy models are used, which receive as input features specified by specialists that also make the development of an annotation system an expensive task. Nonetheless, recurrent neural sequence-to-sequence network models (Sutskever, Vinyals, and Le 2014) have recently emerged as a possibility of creating annotation systems that do not require the specification of features to serve as input to the system. This paper aims to address the issue of automatic semantic annotation, particularly the association of ontological categories to lexical items, through the construction of an annotator based on a deep neural network. In our application the vocabulary Schema.org, founded by Google, Microsoft, Yahoo and Yandex, was used as the top-level ontology which provided the concepts for the annotation. The results showed that this approach can obtain better results than the features-based annotation models.

This paper is organized as follows. The next section gives an overview of the works that have a relation with this research, presenting the advances and highlighting the points that can be improved. Section describes the *corpus* used in the tests as well as the top-level ontology that provided the classes that served as labels. Section presents the neural model architecture used to perform the annotation process. The results achieved in the classification stage and a discussion about them are discussed on the section and finally the conclusions are presented at section.

Related works

In this section some work is described that have an approach that is related to the one adopted in the present work.

(Sukhareva and Chiarcos 2015) proposed an annotator based on ontologies for the POS (part-of-speech) labeling using a heterogeneous database (corpora that have different annotations). Although the authors trained the network with corpora with different sets of *tags* and degree of granularity, the annotations were partially compatible. The algorithm used in the research was a neural network with Rprop (Resilient back-propagation), and the classes for language categories were specified by OLiA (Ontologies of Linguistic Annotation). The authors' intention was to execute automatic POS tagging using the morphosyntactic categories present in OLiA, differing from our approach, since the ontology used in our work provided concepts observed in Web sites, that is, they reflect what is observed on the Web.

(Chiu and Nichols 2016) proposed the use of a hybrid bidirectional LSTM and CNN neural network for Named

Entity Recognition (NER) task. NER is a type of semantic annotation restricted to a small number of classes, such as Person, Organization and Location. Hence, the techniques that work for NER can work for semantic annotation for a larger number of classes. They used as input only tokenized text and publicly available word embedding. The resulting system proved to be competitive on the CoNLL-2003 dataset and surpassed the previously reported state of the art performance on the OntoNotes 5.0 dataset by 2.13 F1 points. In another test they used two lexicons constructed from publicly-available sources, and their system established new state of the art performance with an F1 score of 91.62 on CoNLL-2003 and 86.28 on OntoNotes. This work has a strong intersection with our work since it uses a bidirectional LSTM neural network to tackle the semantic annotation problem. The difference is that our task encompasses a larger number of classes and, as will be seen, we get strong results with a rather simpler network.

(Mendonça-Júnior, Barbosa, and Macedo 2016) also used a bidirectional LSTM and CNN neural network to perform NER as in the previously mentioned work. The main difference is that they applied the system on various corpora in Brazilian Portuguese. The model also obtained superior performance to traditional classifier models, such as Conditional Random Fields. Backing up the hypothesis that Deep Neural Networks are currently the best approach for this type of task.

In the work proposed by (Andrade 2018) is presented a semantic annotator that uses the top level categories of the Schema.org ontology as labels. He used as a classification model the discriminative technique of Conditional Random Fields (CRF). CRF is a technique that requires a feature engineering work and has high memory and processing requirements, but produces good classification results, especially when it is necessary to relate features that occur far apart in the input sequence. The system proposed yielded excellent prediction results, achieving results above 85% in the F1-score for all classes and a general average of 93.5%. The objectives and dataset of this work are the same of our work, and the distinction is in the technique used. Thus, the results of this work were used as a measure of comparison with our results.

The Ontology and the Corpus

Schema.org ©[1] was the ontology selected for this project for being based on evidence from corpora and for being supported by big technology companies. Indeed, as stated on its site, Schemas.org is a vocabulary that covers entities, relationships between entities and actions. The initiative is founded by Google, Microsoft, Yahoo and Yandex, and the vocabulary is developed by an open community process. This ontology is the result of a joint effort to improve the quality of the web, being a structured data marking scheme that is supported by the main search engines. The core vocabulary currently consists of 598 classes, 862 Properties, and 114 Enumeration values.

[1]https://schema.org/

```
<s><tok base="jame" msd="NNP" ne="PERSON" schema="PERSON"> James</tok>
<tok base="t." msd="NNP" ne="PERSON" schema="PERSON">T.</tok> <tok
base="moore" msd="NNP" ne="PERSON" schema="PERSON">Moore</tok> <tok
affix="ed" base="be" msd="VBD" ne="O" schema="O">was</tok> <tok affix="ing"
base="work" msd="VBG" ne="O" schema="ACTION">working</tok> <tok base=";"
msd=";" ne="O" schema="O">;</tok><tok base="special" msd="JJ" ne="O"
schema="O">special</tok><tok base="interest" msd="NN" ne="O" schema=
"INTANGIBLE"> interest</tok>
```

Figure 1: A segment of the *corpus* annotated with Schema.

For this project, the top-level concepts were used because, according to (Guarino 1998) *Top-level ontologies describe very general concepts like space, time, matter, object,event, action, etc., which are independent of a particular problem or domain.* The categories present in the top level of Schema.org which were used as labels in our research, being the following: *Action*, *Creative_Work*, *Event*, *Intangible*, *Organization*, *Person*, *Place* and *Product*.

Each category has a formal definition, which is key to correctly assign a lexical item to one of the categories. Moreover, the fact that it is an ontology based on corpora evidence, concepts are easier to understand than the ones from an ontology based on philosophical concepts.

In this project we used two corpora: OANC (Open American National Corpus) corpus (Fillmore et al. 1998) that was used in tests 1 and 2, and Wikiner (Nothman et al. 2013), which was used in test 3. Test 3 was performed with a set of different data to demonstrate the ability of our model to generate good results in different corpora. Both corpora used will be described in more detail later in this article.

The first dataset chosen to test the proposed model was the OANC (Open American National Corpus) corpus (Fillmore et al. 1998) (Ide 2013). The choice of dataset occurred because of its size and the wide range of textual genres. The data volume ensures that the algorithm can capture the co-occurrence and relations between words. The corpus is available online free of charge at the url http://www.anc.org, and contains about 15 million words of contemporary American English for a variety of topics. An important feature is that the OANC data has the following annotations: Structural markup (sections, chapters, etc.) down to the level of paragraph; Sentence boundaries; Words (tokens) with part of speech annotations and lemma using the Penn tagset; Noun chunks; Verb chunks; and Named Entities (Person, Location, Organization, Date). These annotations make the dataset ideal for use in supervised learning, one more factor that was considered in its choice.

Due to the computational costs to annotate a large volume of documents, it was necessary to select a corpus fragment to make project development manageable in a timely manner. The fragment extracted from the corpus for training and testing the neural network was supplementary annotated with the top level tags of Schema.org by (Andrade 2018). It was also necessary to normalize the data by excluding irrelevant information such as paragraph marks, blanks and other structural marks. The Figure 1 shows a segment of the annotated corpus. After normalization, we obtained the resulting corpus presented the numbers shown in Table 1.

The texts used as input to the model were in plain Unicode format, segmented into sentences where each word receives

Table 1: OANC in numbers.

Items	Quantity
Input Sentences	3,295,069
Vocabulary size	222,553
Sentence size	50

According|VBG|O to|TO|O Peter|NNP|I-PER Kropotkin|NNP|I-PER students|NNS|O from|IN|O the|DT|O Massachusetts|NNP|I-ORG Institute|NNP|I-ORG of|IN|I-ORG Technology|NNP|I-ORG include|VBP|O the|DT|O US|NNP|I-LOC member|NN|O of|IN|O the|DT|O French|NNP|I-ORG Academy|NNP|I-ORG of|IN|I-ORG Sciences|NNPS|I-ORG

Figure 2: Wikiner sentence sample

a tag and where such tags are assigned according to the associated ontological class: *Action*, *Creative_Work*, *Event*, *Intangible*, *Organization*, *Person*, *Place*, *Product*, and *Other* for cases in which the word does not belong to any of the aforementioned classes.

The second *corpus* used was Wikiner, which is the result of (Nothman et al. 2013) work. Wikiner is a silver standard annotated corpus for named entity recognition. It is multilanguage, consisting of texts written in several genres extracted from Wikipedia. In the test performed, the portion available in English was selected.

The corpus has one sentence per line, each token is separated by a blank space and contains three items: a text token, a POS tag, and a Beginning-Inside-Outside (BIO) tag followed by the token class acronym, as shown in the figure 2. It was necessary to normalize the corpus by removing markers, spaces and also the initials of the BIO pattern of the classes, leaving them in a leaner format.

It is noteworthy that Wikiner was annotated with five ontological classes: *Organization*, *Person*, *Place*, *Miscellaneous (Misc)* and Other. The *misc* class encompasses words marked as event, action, product, intellectual production, and intangible things of the Oanc *corpus*.

Table 2: Wikiner in numbers.

Items	Quantity
Input Sentences	142,153
Vocabulary size	7,875
Sentence size	50

Proposed Model

After defining the dataset and the tags that would be used, we set out to prepare the model and prepare the data to feed the network. The entire dataset was converted to vector form using the GloVe algorithm proposed by (Pennington, Socher, and Manning 2014). They used aggregated global word-word co-occurrence statistics from a corpus to train the algorithm, allowing it to capture linear substructures of the word vector space. Converting the words in the dataset to a vector of float point numbers is an essential step when one uses neural network models, once it makes it possible code the sentences through an embedding layer. In our test we used

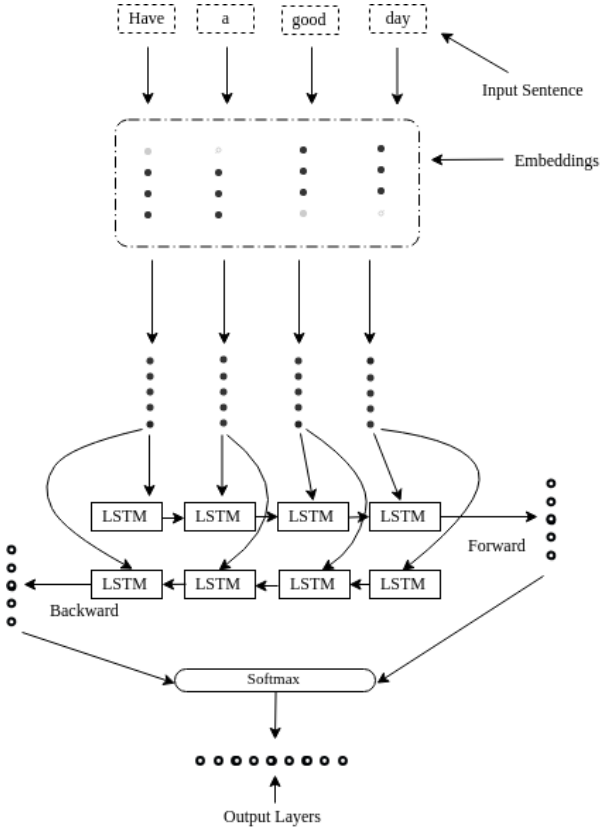

Figure 3: The proposed network architecture.

vectors with 200 elements, and this number was obtained from empirical tests that showed that vectors with greater number of elements did not bring significant improvements.

The neural network implemented was of the type *Bidirectional Long-Short Term Memory* (BiLSTM). The LSTM network can handle arbitrary-sized sentences, detecting relationships between distant words in the sentence. Being bidirectional allows these relationships to be detected both in the previous words, and in the words subsequent to the current word. Several architectural arrangements have been attempted and Figure 3 shows the architecture of the network that returned the best results.

The first layer of the network is an embedding layer that receives the sentences in the form of sequence numbers and substitutes each word for its vectored representation. As previously mentioned, we used 200 size vectors resulting from training on the whole corpus with the Glove algorithm. The embedding layer receives each sentence in the form of integer indices and encodes them using the Glove matrix, passing the result to the next layer. The subsequent layer is a Bi-LSTM layer with 32 hidden elements. This layer is responsible for learning which annotation sequence should be associated with the input sentence. We define this layer to return the entire sequence and not just the result of the last time step. The remaining hyper-parameters were left with the default values. Finally, the last layer of the model is a dense layer enveloped with a time distribution wrapper that

according to the author of Keras, François Chollet (Chollet and others 2015), "applies a same Dense (fully-connected) operation to every time step of a 3D tensor". The dense layer uses a softmax function as an activation function in order to approximate the probability of each of the ontological classes.

The model Hyper-parameters that were established empirically and returned the best results. Table 3 presents the main hyper-parameters used in the construction of our model.

Table 3: The main model Hyper-parameters.

Hyper-parameters	Values
Word Embedding Vector Size	200
Batch Size	32
Dense Layers elements	9
Activation function	Softmax
Loss function	Categorical crossentropy
Optimizer	Adam
Number of epochs	50

A Batch Size of 32 was used to allow the speed of convergence, but at the same time, preventing the model from being stuck in a local minima. The activation function in the LSTM layer was hyperbolic tangent which is the standard, and in the final dense layer the Softmax function was used, which is the default activation function when the problem is of the multi-class classification type, which is the case. Softmax assigns decimal probabilities to each class in a multi-class problem where those decimal probabilities must add up to 1.0, generating a probability distribution over the nine different possible outcomes. Each score will be the probability that the word belongs to one of our 9 ontological classes.

The loss function chosen was the categorical crossentropy (equation 1), being the one suggested for a many-class classification problem. It minimizes the distance between the probability distributions output by the network and the true distribution of the targets (Chollet and others 2015).

$$\mathcal{L}(y, \hat{y}) = -\sum_{n}\sum_{i} y_i^{(n)} \log \hat{y}_i^{(n)} \tag{1}$$

where:

y *is the ground-truth class probabilities.*
$\hat{y}$ *is the model predicted probability distribution.*
i *is the class index.*
n *is the sample index.*

The optimization algorithm chosen for our deep learning model was the Adam optimizer (Adaptive Moment Estimation) (Kingma and Ba 2014). According to the authors Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. It computes adaptive learning rates for each parameter and also keeps an exponentially decaying average of past gradients, similar to momentum approach used in other stochastic gradient descent optimization algorithms. The authors state that the algorithm is robust and well-suited to a wide range of non-convex optimization problems in the field machine learning.

Results

The LSTM network was implemented through the Keras framework (version 1.2.0) (Chollet and others 2015) which, in turn, ran at the top of the Tensorflow framework(Abadi et al. 2016). Keras is a high level framework that allows the construction of neural networks in an easy way through the python language where the user can gradually specify the layers of the networks or define their functional relations. The implemented model ran on a computer with an Intel I7 processor of eighth generation and 16 GB RAM, equipped with an NVIDIA GeForce GTX 680 graphics card. The operating system used was Linux Mint 18.3.

There were three types of tests performed whose characteristics are detailed below:

Test 1: used only a short section of the corpus to validate the neural network implemented. In this test we worked with only 4 classes, the only ones present in the selected section (*Action*, *Organization*, *Person* and *Other*). The performance evaluation for test 1 used the *accuracy* metric and *F1-score* (Equation 4), both metrics presented similar values of precision. 1000 sentences were selected using the split 80% for training and 20% for testing. The purpose of this experiment was to verify the performance of the implemented model, evaluate and adjust the established parameters.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Test 2: characterized by the total use of the normalized corpus. More than 3 million sentences were consumed by bi-LSTM, using as tags the 9 ontological classes mentioned previously. This test was the basis for verifying the performance of bi-LSTM with a greater number of data. Based on test 1, the metric used at this stage was the *F1-score* and the number of epochs was 50.

Taking into account the second test and selecting the best result achieved for the assignment of the ontological classes, we constructed the Confusion Matrix shown in the Table 4.

Observing Table 4 it is possible to note that the class with the highest absolute numbers of spurious classification is, as expected, the class *Other*, due to the large number of elements. It is also the class that receives the highest number of erroneous classifications of the other classes, being the exception the class *Event* that has the highest number of erroneous classifications assigned to the class *Action*. Mistaken *Event* for *Action* is something that can be expected since there is a semantic proximity (actions generate events). A clear disadvantage of the dataset is that it is unbalanced, which can make it difficult to learn the relations related to the identification of a particular class. Table 5 shows the accuracy for each class collected from the test with the best result.

Table 4: Confusion matrix

	Action	Organization	Person	Product	Creat.	Place	Event	Intangible	Other
Action	**32097**	49	124	1	6	36	7	9	1047
Organization	6	**30241**	735	44	1	962	0	3	5225
Person	2	727	**40216**	8	2	392	0	0	3115
Product	0	38	25	**8456**	3	1	0	1	41
Creative_Work	2	1	2	2	**10771**	2	0	0	67
Place	6	1285	786	3	0	**23573**	0	1	2615
Event	12	2	0	0	0	0	**2713**	0	5
Intangible	13	3	5	1	1	1	0	**16728**	305
Other	746	8291	6955	24	6	2798	5	220	**2086321**

Table 5: Results per class

	precision	recall	F1-score
Action	0.98	0.96	0.97
Organization	0.74	0.81	0.78
Person	0.82	0.90	0.86
Product	0.99	0.99	0.99
Creative_Work	1.00	0.99	1.00
Place	0.85	0.83	0.84
Event	1.00	0.99	0.99
Intangible	0.99	0.98	0.98
Other	0.99	0.99	0.99
avg / total	**0.98**	**0.98**	**0.98**

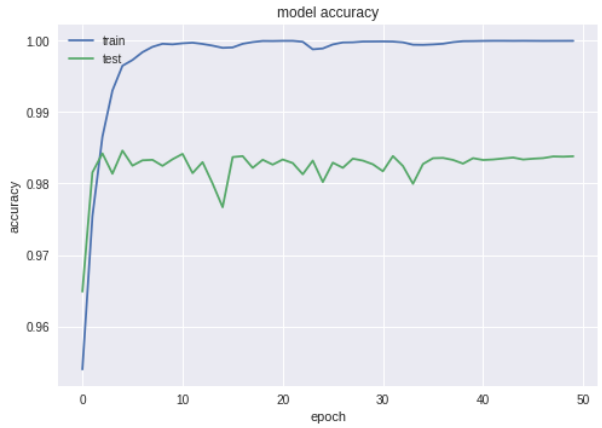

Figure 4: The model accuracy curves.

The model presented an accuracy average over all classes of 98% in the F1-score. Each ontological class was evaluated individually and the values obtained in the Precision, Recall, F1-score, and Support. The results achieved are quite impressive. The class that got the lowest F1-score was *Organization*, achieving 78%. Six out of the nine classes received an F1-score greater than 96%.

In the Test 3 scenario the *corpus* used was the Wikiner, employing around 150,000 sentences, as described in the section . The main objective of this experiment was to validate the developed model, comparing its results with another model that used the same corpus. The results obtained with this test were compared with those obtained by (Rondeau and Su 2016) whose F1-score reached 89.28% overall precision. The model proposed by (Rondeau and Su 2016) consisted of a NeuroCRF network performing in conjunction with a Bi-LSTM.

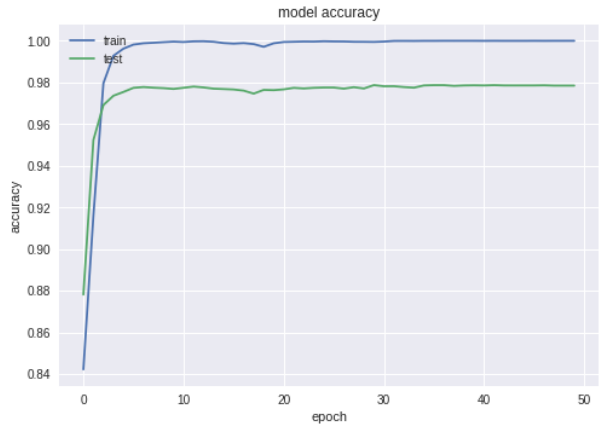

Figure 5: The model accuracy curves.

As can be seen in Figure 5, in this test the training and test datasets exhibited similar precision curves. The third test is basically a superficial remodeling of the second and, according to table 6, the Misc class presented the highest number of false negatives due to the variety of entities involved in the same class.

In the table 7 the results of the test performed are shown and it can be concluded that the Bi-LSTM network generated better results than those obtained by the authors mentioned previously. Four of the five classes evaluated showed results greater than 89%, presenting an overall F1-score of 95%.

Conclusions

Our main contribution is to achieve good results in the assignment of ontological category labels to lexical items through a rather simple bidirectional LSTM neural network model. In addition to not needing a feature engineering phase, as is the normal case of neural network models, we didn't use a hybrid model with convolutional networks, and we did not use embedding at the character level, as was the case in other related works. Nonetheless, our results were much better than the work presented by (Andrade 2018) that used a CRF model. The non-use of character-level embed-

Table 6: Confusion matrix

	Organization	Place	Misc	Person	Other
Organization	**5800**	13	12	10	450
Place	130	**4880**	80	30	170
Misc	20	40	**3250**	80	790
Person	60	20	10	**4390**	310
Other	60	59	150	60	**102690**

Table 7: Results per class

	precision	recall	F1-score
Organization	0.91	0.87	0.90
Place	0.95	0.92	0.95
Misc	0.87	0.78	0.84
Person	0.96	0.90	0.93
Other	0.98	1.00	0.99
avg / total	**0.97**	**0.93**	**0.95**

ding is due to the hypothesis that the ontological semantic annotation does not depend so much on features at character level, but rather on features related to word co-occurrence. The results corroborate the thesis that recurrent neural networks models, especially the variations of LSTM and GRU, are the current choice for the processing of sequences processing in the scope of NLP.

Following the idea of (Firth 1957) stating that *you shall know a word by the company it keeps*, one approach that may improve the labelling scores would be to feed the neural network with constructions structures (Goldberg 2006) that capture the usual schemes of relationships between words. This proposal is the path we must pursue in future research.

Acknowledgements. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and also by the funding agencies FAPEMIG and CNPq.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, 265–283.

Andrade, G. C. 2018. Semantic enrichment of american english corpora through automatic semantic annotation based on top-level ontologies using the crf classification model. Master's thesis, Universidade Federal de Viçosa, Brazil.

Chiu, J. P., and Nichols, E. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics* 4:357–370.

Chollet, F., et al. 2015. Keras.

Fillmore, C.; Ide, N.; Jurafsky, D.; and Macleod, C. 1998. An american national corpus: A proposal. In *Proceedings of the First Annual Conference on Language Resources and Evaluation*, 965–969.

Firth, J. R. 1957. *Studies in linguistic analysis*. Wiley-Blackwell.

Goldberg, A. E. 2006. *Constructions at work: The nature of generalization in language*. Oxford University Press on Demand.

Guarino, N. 1998. Formal ontology in information systems. In *Proceedings of the first international conference (FOIS'98)*, volume 46. Trento, Italy: IOS press.

Ide, N. 2013. An open linguistic infrastructure for annotated corpora. In *The People's Web Meets NLP*. Springer. 265–285.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mendonça-Júnior, C. A. E.; Barbosa, L. A.; and Macedo, H. T. 2016. Uma arquitetura híbrida lstm-cnn para reconhecimento de entidades nomeadas em textos naturais em língua portuguesa. In *Proceedings of XIII Encontro Nacional de Inteligência Artificial e Computacional*, 241–252.

Nothman, J.; Ringland, N.; Radford, W.; Murphy, T.; and Curran, J. R. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence* 194:151–175.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Pustejovsky, J., and Stubbs, A. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. "O'Reilly Media, Inc.".

Rondeau, M.-A., and Su, Y. 2016. Lstm-based neurocrfs for named entity recognition. In *INTERSPEECH*, 665–669.

Sukhareva, M., and Chiarcos, C. 2015. An ontology-based approach to automatic part-of-speech tagging using heterogeneously annotated corpora. In *Proceedings of the Second Workshop on Natural Language Processing and Linked Open Data*, 23–32.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc. 3104–3112.