Spatial Role Labeling System Capturing Both Characters and Word Information Using BiLSTM and CRF

Alaeddine Moussa[1,2,3], Sébastien Fournier[1], Khaoula Mahmoudi[3], Bernard Espinasse[1], and Sami Faiz[3]

[1] Aix Marseille Université, CNRS, LIS UMR 7020
[2] Univ. Manouba, ENSI, Campus Universitaire Manouba
[3] Université de Tunis El Manar, ENIT, Laboratoire de Télédétection et Systèmes d'Information à Référence Spatiale (UR-TSIRS-ENIT)

Abstract. Spatial role labeling (SpRL) is a complex task that requires large amounts of feature engineering to achieve high performance. This paper proposes an end-to-end neural network architecture for SpRL that automatically captures both words and characters-level information. Our proposed model combines a bidirectional LSTM network with CharLSTM and Conditional Random Field (CRF) Layer (CharLSTM-BiLSTM-CRF). Our system outperforms all classical baseline models and is in line with those using deep learning approaches.

Keywords: Spatial Role Labeling · Spatial relation extraction · Deep learning.

1 INTRODUCTION

In natural language processing, Semantic Role Labelling (SRL) assigns labels to words or phrases in a sentence that indicate their semantic role in the sentence, such as agent, goal, or outcome.

Spatial Role Labeling (SpRL) is the process of extracting spatial information between objects in a text. Linguistic constructs may communicate complex relationship structures and spatial relationships between them and movement patterns across space relative to a reference point. A spatial relationship between two objects is typically expressed in natural language through a preposition (e.g. in, on, at, ...) or prepositional expression (on top of, inside of, ...). SpRL is used in many application areas such as robotics, maritime navigation, traffic management, and query response systems.

As SpRL is a cumbersome and complex task, early consideration was given to automating it, mainly through machine learning techniques, which are based on artificial intelligence. Implementing these techniques led to the development of specific systems, using different approaches based on different types of machine learning, namely symbolic, statistical, and deep learning.

We hypothesized that character embeddings combined with word embeddings and conditional random field (CRF) could improve SpRL tasks' accuracy.

Indeed, character embeddings help syntactic tasks in nature like SpRL, as the character patterns within words are strongly related to their syntactic function [11]. Furthermore, adding a CRF in top of a bidirectional long-short term memory (BiLSTM) architecture improved the performance of sequence labeling tasks like NER [3].

In this paper, we propose an end-to-end neural network approach to resolve the SpRL task. We first use character embeddings and long-short term memory (LSTM) to encode every word as a vector. Then we concatenate character- and word-level information and feed them into bi-LSTM. Finally, we use a conditional random field CRF to decode tags for the entire sentence. The rest of the document is organized as follows. Section 2 describes the fundamental concepts of SpRL. Section 3 presents the systems involved in SpRL. Section 4 describes the layers of our neural network architecture. Section 5 outlines the experimental evaluation of the adapted architecture. Finally, in Section 6, we present our conclusions and discuss possible directions for future work.

2 Spatial Role Labeling

The SpRL task [9] implies automatic recognizing of spatial information in a sentence with a set of spatial role tags. Let us take the sentence:

$$[Trees]_{TRAJECTOR}\ [in]_{SPATIALINDICATOR}\ the\ [background]_{LANDMARK}\ .$$

where the token "Trees" is a TRAJECTOR (TR), "background" a LANDMARK (LM), and "in" the SPATIAL INDICATOR (SI). A trajector describes a central object of a spatial scene. It can be static or dynamic; including persons, objects, or events. The landmark is a spatial role label that denotes the location or the motion of the trajector. It indicates a secondary object of a spatial scene, to which a possible spatial relation can figure out. Spatial Indicator is a spatial role label allocated to a word or a phrase to flags a spatial relation between objects (trajectors and landmarks).

Many studies have focused on SpRL, developed at the crossroads of the geomatics and natural language processing (NLP) communities within SemEval campaigns. SemEval is a series of ongoing evaluations of computational semantic analysis systems, aimed at exploring the nature of meaning in language. Although meaning is intuitive to humans, the transfer of these intuitions to computer analysis has proven to be complicated.

The SemEval campaigns in automatic information extraction, of the years 2012, 2013, and 2015, involved challenges to be taken up in SpRL on a specific proposed corpus of reference texts. Thus the SpRL task at SemEval 2012 [7] focused on the roles of trajectors, landmarks, spatial indicators, and the links between these roles that form spatial relations.

In the SemEval 2013 SpRL task [6], introduced new roles to include motion indicators, paths, directions, and distances to capture the fine-grained spatial semantics of static spatial relationships and also to take into account dynamic spatial relationships.

The SpaceEval task, introduced in SemEval-2015, adopted a more advanced annotation specification with respect to ISOspace [15]. SpaceEval, first, enriches the semantics' granularity in both static and dynamic spatial configurations, and secondly, by extending the variety of annotated data and the domains considered. Indeed, the concept of place is distinguished from the concept of a spatial entity.

3 SpRL Systems

The Systems covered in this section focus only on SemEval 2012 and 2013, and do not explore SpaceEval 2015 campaign considering the change of corpus annotation.

The KUL-SKIP-CHAIN-CRF [10] system adopted a pipeline approach. In the beginning, spatial indicators are found and labeled for each sentence. Then, given a spatial indicator, the second task consists of classifying parts of an input sentence with landmark or trajector labels. To identify the spatial indicator, authors use an external corpus The Preposition Project (TPP) labeled to learn the sens of preposition through a set of linguistic features (lemma, POS,...). As a result, the binary classification of a preposition's spatial is made. In the same way, Trajectory and Landmark are identified to leverage many linguistically motivated features. Finally, spatial indicators, trajectors, and landmarks are combined into spatial relation triplets.

The UTDSpRL [17] system used a join approach to tackle the SpRL task. Authors expose that the pipeline approach cannot perform well if a spatial roles arguments are considered in isolation. They propose an alternative system that jointly decides whether a given candidate triple expresses a spatial relation or not. The approach used a recall heuristics to find spatial relation candidate triple. A hand-crafted dictionary was used to detect SPATIAL_ INDICATOR candidates, allowing the decrease of negative relation candidates in comparison to the pipeline method, which considers every phrase as a spatial role. Then, noun phrase heads were treated as TRAJECTOR and LANDMARK candidates. Finally, a Support Vector Machine (SVM), was trained with some manually engineered features chosen by an automatic feature selector, to classify the relation candidate.

The UNITOR-HMM-TK [1] system uses a sequence-based classifier. First, classifies spatial and motion indicators, then built on these outcomes to identify spatial roles through a set of lexical and grammatical features. The generated candidate spatial relations are verified by a Support Vector Machine (SVM). This approach's significant contribution is to adopt smooth grammatical features instead of a full syntax of the sentence.

SpRL-CNN system [14] rely on an adapted version of the nlpnet[4] system. First, a spatial indicator is identified by a multilayer perceptron (MLP) witch it

[4] http://nilc.icmc.usp.br/nlpnet/

converts the tokens (spatial indicator) into feature vectors. Then, these vectors are fed to Conventional Neural Network (CNN) to classify the argument and spatial relation triples. Additionally, pre-trained Glove word embeddings were used to reduce the impact of words not seen in the training data.

The VIEW system [12] consists of producing word embedding vectors from the Microsoft COCO dataset to enrich the generated "Feature" by SpRL methods. "Microsoft COCO dataset" represents a collection of segmented images. At first, the systems start by transforming each image's legends into word embedding vectors that will be fed into a Long Short Term Memory (LSTM) network. The goal is to be able to encode the visual information of images within the "Feature vector" produced by [8]) (generating from POS, semantic roles, syntactic dependencies, ...) by concatenating them with the "embedding" of VIEW.

Most of the existing systems depend heavily on manual features and do not benefit from the advances of NLP systems based on deep learning techniques.

4 Proposed Architecture

In this section, we provide details about our neural network architecture (Figure 1) inspired by [13]. We introduce the neural components in our network one-by-one upward.

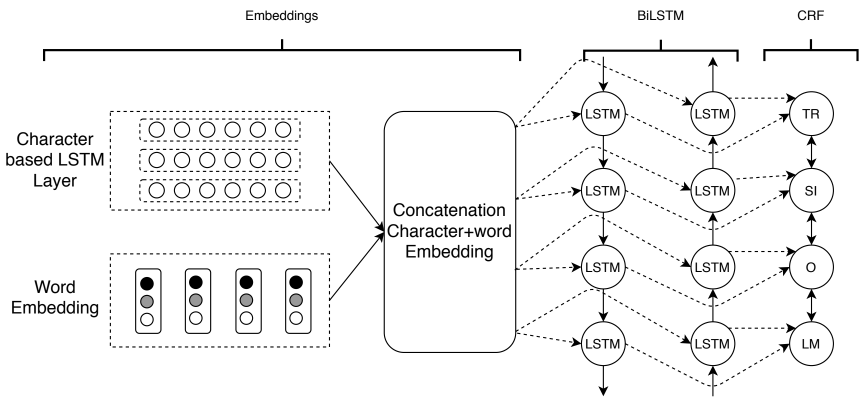


Fig. 1: Global architecture of our neural network

4.1 Character-level Representation

Character-level contains much information about the meaning of the word. It proves than it can memorize the arbitrary aspects of word spelling. Using this information can significantly improve the quality of the model [2]. To encode the structure of words, we will use an LSTM. In practice, we will encode each character into a vector using one-hot encoding and feed them into an LSTM one at a time.

4.2 Bi-directional LSTM

LSTM The Long Short-Term Memory (LSTM) architecture is a concrete instantiation of the abstract RNN architecture. In practice, RNNs fails to learn long-distance dependencies due to the vanishing gradients problem. Indeed, errors detected in the later steps in the sequence decrease quickly in the back-propagation phase, and do not attempt the first input signals. LSTM was designed to address this shortcoming using a controllable gating mechanism with a memory cell. Differentiable gating mechanisms determine, at each stage, which inputs parts to write in memory, and which memory's parts are overwritten. The LSTM architecture is described in mathematical terms for a unit at time j as:

$$c_j = f \odot c_{j-1} + i \odot z \tag{1}$$

$$h_j = o \odot tanh(c_j) \tag{2}$$

$$i = \sigma(x_j W^{xi} + h_{j-1} U^{hi}) \tag{3}$$

$$f = \sigma(x_j W^{xf} + h_{j-1} U^{hf}) \tag{4}$$

$$o = \sigma(x_j W^{xo} + h_{j-1} U^{ho}) \tag{5}$$

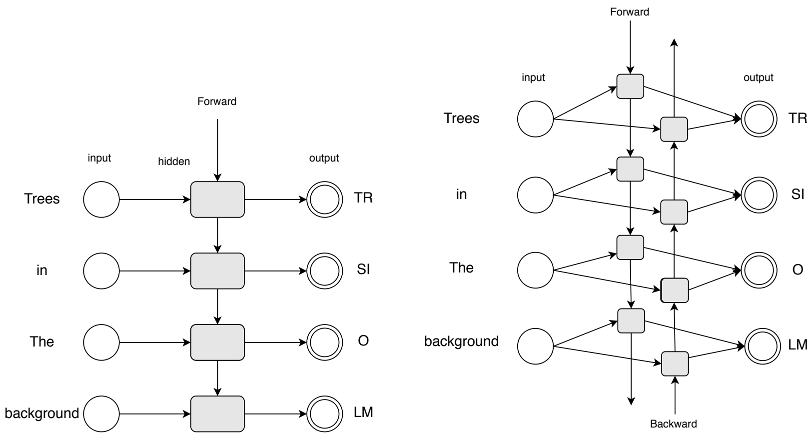
$$z = tanh(x_j W^{xz} + h_{j-1} U^{hz}) \tag{6}$$

where $\boldsymbol{c_j}$ denotes the memory cell and $\boldsymbol{h_j}$ is the hidden state. σ denotes the element-wise sigmoid function, and $\odot$ is the element-wise product. $\boldsymbol{i}$, $\boldsymbol{f}$, and $\boldsymbol{o}$ are the three gates. $\boldsymbol{W_{xi}}$, $\boldsymbol{W_{xf}}$, $\boldsymbol{W_{xc}}$, $\boldsymbol{W_{xo}}$ denote the weight matrices of different gates for input $\boldsymbol{x_j}$, and $\boldsymbol{U_{hi}}$, $\boldsymbol{U_{hf}}$, $\boldsymbol{U_{hc}}$, $\boldsymbol{U_{ho}}$ are the weight matrices for hidden state $\boldsymbol{h_j}$. The values of the gate are determined on the basis of linear combinations of the current input $\boldsymbol{x_j}$ and the previous state $\boldsymbol{h_{j-1}}$ using a sigmoid function. The updated candidate $\boldsymbol{z}$ is estimated as a linear combination of $\boldsymbol{x_j}$ and $\boldsymbol{h_j}$ via a tanh function. Figure2a shows a LSTM sequence tagging model.

Bidirectional LSTM Access to both past (left) and future (right) contexts is useful for many sequence labeling tasks. However, the hidden state h of the LSTM takes knowledge took only the prior words for consideration. The Bidirectional LSTM [5] solves this problem through encoding the ith word in a sequence by the concatenation of two LSTM, one reading the sequence from the beginning, and the other reading it from the end. Figure2b shows a BiLSTM sequence tagging model.

4.3 CRF

When processing sequence labeling tasks, each labeling decision is produced independently of the others words, but in many cases neighboring decisions leverage



(a) LSTM architecture (b) BiLSTM architecture

Fig. 2: LSTM and BiLSTM architectures

each other. The conditional random field (CRF) solves this problem and model label sequence jointly. Given an input sequence (word)

$$x = (x_1, ..., x_n) \tag{7}$$

and a sequence of SpRL labels

$$y = (y_1, ..., y_n) \tag{8}$$

for x. The CRF models the conditional probability as follow:

$$p(y_1, ..., y_n | x_1, ..., x_n) \tag{9}$$

We denotes the set of possible label sequences for x to some d-dimensional feature vector by a feature map:

$$\Phi(x_1, ..., x_n, y_1, ..., y_n) \in \mathbb{R}^d \tag{10}$$

The the probabilistic model for over all possible tag sequences will be define as :

$$p(s|x; w) = \frac{e^{(w.\Phi(x,y))}}{\sum_{x'} e^{(w.\Phi(x',y))'}} \tag{11}$$

where s extends over all possible output sequence and w is the parameter vector.

During training, we use the maximum conditional likelihood estimation. The regularized log-likelihood function L is:

$$L(w, b) = \sum_i logp(y|x; w, b) \tag{12}$$

Finally, we can find the highest conditional probability tag y^* for a sentence x by:

$$y^* = argmaxp(y|x; w, b) \tag{13}$$

4.4 CharRNN-BiLSTM-CRF

We construct our neural network architecture by feeding the BiLSTM output vectors into a CRF layer. For each word, the character-level representation is given by the first LSTM with character embeddings as inputs. Then the character-level representation vector is concatenated with the word embedding vector to feed into the BiLSTM network. Finally, the output vectors of BiLSTM are fed to the CRF layer, taking advantage of neighboring tag information in predicting current tags. This network can efficiently use past input features via the LSTM layer and sentence level tag information via the CRF layer, as shown in Figure 1.

Indeed, from equation 11 the expression

$$w.\Phi(x, y) = score_{crf}(x, y) \tag{14}$$

represents the potential functions, also known as scoring function, that indicates how y fits x. In the CRF layer, we aim to replace the linear scoring function by a non-linear neural network. We define the score by:

$$score_{BiLSTM-crf}(x, y) = \sum_{i=0}^{n} W_{y_{i-1},y_i}.BiLSTM(x)_i + b_{y_{i-1},y_i} \tag{15}$$

With the score function constructed, we can efficiently use past and future tags to predict the current tag and optimize the conditional probability $p(y|x; w, b)$ and propagating back through the network.

5 Experiments

This section presents the experiments in SpRL, using various neural network architecture; MLP, LSTM, LSTM+CRF, BiLSTM+CRF, and CharRNN+BiLSTM+CRF. First, we present the SpRL corpus and then the results obtained with these two models. Finally, we compare these results and try to interpret them.

5.1 SpRL corpus

SemEval-2012 Data The first corpus is a subset of the IAPR TC-12 image Benchmark [4]. It contains 613 text files that include 1213 sentences in total. The original corpus was available without copyright restrictions [5]. Tourists took 20,000 images with textual descriptions in different languages. The texts describe objects in a scene together with their absolute and relative positions in the image. The data is released in XML format.

[5] http://www.cs.tulane.edu/ pkordjam/SpRL.htm# data

SemEval-2013 Data The dataset for SemEval2013 includes two distinctive corpora. The first one is the SemEval2012 Data presented in the previous subsection but with some modifications. The Data was transformed into a span-based annotation, and some annotation error was corrected.

The second corpus originates from the Confluence Project [6]. This project tries to outline all possible latitude-longitude intersections on earth, and people who visit these intersections provide written narratives of the visit. The authors use the same annotation then IAPR TC-12 image extended with "motion_ indicator", "path", "direction," and "distance" tags to denotes the dynamic spatial aspect.

Systems were evaluated mainly into two tasks : Individual role identification and spatial relation extraction. For SemEval2013 task 3, all reported results followed the relaxed evaluation criteria as motioned by the organizers.

5.2 Sequential labeling for Spatial Role Classification

The task of sequential labeling for SpRL is to assign a spatial role to a specific word in a sentence. We convert the original XML corpus to IOB2 format (beginning (B), the inside (I), or ending (O)). Thus, the labeling of sentence 2 for SpRL-2012 data can be as follows:

$$\text{Trees}/_{\textsc{b-tr}}\ \text{in}/_{\textsc{b-si}}\ \text{the}_{/\textsc{o}}\ \text{background}/_{\textsc{b-lm}}$$

.

Besides for SpRL-2013 it will be:

$$\text{Trees}/_{\textsc{b-tr}}\ \text{in}/_{\textsc{b-si}}\ \text{the}_{/\textsc{b-lm}}\ \text{background}/_{\textsc{i-lm}}$$

To test the efficiency of our system, we conducted an ablation test. Several combinations of neural network architectures were used, namely LSTM + CRF, BiLSTM+CRF, and char+BiLSTM+CRF. These experiments shown in tables 1 and 2 allowed us to highlight the efficiency of BiLSTM compared to LSTM but also to confirm the result of the previous work that the contribution of embedding character-level embedding in the sequence labeling problem. Finally, the CRF layer allows to optimize further the performance of our model.

5.3 Relation identification

For the relation extraction task, we aim to figure out which spatial roles discovered in the previous classification phase, can be composed as valid spatial relations. Inspired by [17] where all possible spatial roles are first generated by heuristics and then combinatorially combined to acquire candidate relationships. The network has three input vectors, each referring to the trajector, spatial indicator, and landmark. results for static relation are reported in Table 3.

[6] http://confluence.org

Table 1: Results of various neural architecture for the SemEval 2012 tasks

Network architecture	Label	Precision	Recall	F1-score
LSTM-CRF	SI	0.755	0.865	0.806
	LM	0.676	0.583	0.626
	TR	0.479	0.591	0.529
BiLSTM-CRF	SI	0.917	0.862	0.889
	LM	0.600	0.720	0.655
	TR	0.608	0.687	0.645
Char-BiLSTM	SI	0.871	0.914	0.892
	LM	0.541	0.748	0.628
	TR	0.637	0.586	0.610
Char-BiLSTM-CRF	SI	0.899	0.919	**0.909**
	LM	0.561	0.809	**0.662**
	TR	0.658	0.677	**0.667**

Table 2: Results of various neural architecture for the SemEval 2013 tasks

Network Architecture	Label	Precision	Recall	F1-score
LSTM-CRF	SI	0.805	0.732	0.767
	LM	0.605	0.722	0.658
	TR	0.521	0.553	0.536
BiLSTM-CRF	SI	0.868	0.879	0.873
	LM	0.600	0.720	0.655
	TR	0.589	0.636	0.612
Char-BiLSTM	SI	0.836	0.920	0.876
	LM	0.507	0.705	0.590
	TR	0.620	0.631	0.626
Char-BiLSTM-CRF	SI	0.927	0.879	**0.902**
	LM	0.566	0.789	**0.660**
	TR	0.677	0.606	**0.639**

Table 3: Results of relation extraction for the SemEval 2012 and 2013 tasks

	Relation Extraction	Precision	Recall	F1-score
SemEval 2012	Relation	0.558	0.785	0.652
SemEval 2013	Relation	0.492	0.706	0.580

5.4 Network training procedure

In this section, we provide details about training the neural network. We implement the neural network using the Keras library. We choose the stochastic gradient descent (SGD) method to train our model and update parameters on every training batch. For the BiLSTM network, after empirical tests, we find the best results with a batch size of 32, dropout of 0.1, and 200 LSTM units. The CRF was implemented with the keras-contrib package. For character embedding, we choose to set the dimension to 10 since the longest word in the corpus has 10 characters. We use the Google's Word2Vec with varying the dimension of the embedding matrix. The best result came empirically with dimension vector of 200. Besides, we implemented the Early Stopping mechanism to stop training once the performance of the model began to decrease. According to our experiments, the "best" epoch number appears at around 70 epochs. We also add a five epoch delay to the trigger on which we would like to see no improvement of the model.

5.5 Discussion

The Tables 4 and 5 compares the results of our system with other baselines approaches. We notice that it is difficult to compare individual spatial role classification with [17] because their results are a derivative of a joint classification and not the performance of an individual classifier.

Moreover, UTDSpRL [17] uses a fixed lexicon of spatial indicators was built training data, so it will not recognize any relations that use new indicators.

We observed a drop of performance from SpRl-2012 corpus to SpRL-2013 due to the use of full span annotation instead of headwords. Also, a decrease from training data to the test data has been shown. As mentioned by [17] it is due to a more length and complexity of sentences used in test data. To verify this observation, we perform other experiments with shuffled data using the Scikit-learn library. We preserve the same number of sentences in the original train and test corpus. The results show an increase in F1-score by +9.36 on average in SemEval-2012 and by +9.56 in SemEval-2013, which confirms that the original test data is more challenging to handle.

Besides, we try to understand the prediction of our neural network architecture using the LIME algorithm [16]. LIME (Local Interpretable Model-agnostic Explanations) is an explanation technique that attempts to understand the model by perturbing the input data and perceive how the predictions change. Figure 3 shows an example of prediction the word "floor" for the sentence "light and dark brown tiles on the floor .". We can observe that the model relies on the word "floor" itself and the spatial indicator "on" to assign the label landmark LM.

6 Conclusion

In this paper, we proposed a CharRNN-BiLSTM-CRF neural network architecture for Spatial role labeling mixing character-level and word-level representation. It is an end-to-end model relying on no feature engineering. We outperform all classical baselines models and we are in line with those using deep learning approaches. Notwithstanding the satisfying results, there are several potential directions for future work. These include:

Table 4: F1 scores comparison with the state-of-the-art for the SemEval 2012 tasks

System	TR	LM	SI	Relation
UTDSpRL [17]	**0.707**	0.772	0.823	0.573
KUL-SKIP-CHAIN-CRF [10]	0.646	0.756	0.900	0.500
EtoE-IBT-CLCP [8]	0.673	**0.797**	0.869	0.617
Our CharRNN-BiLSTM-CRF	0.667	0.662	**0.909**	**0.652**

Table 5: F1 scores comparison with the state-of-the-art for the SemEval 2013 tasks

System	TR	LM	SI	Relation
UNITOR-HMM-TK [1]	0.682	**0.785**	**0.926**	0.458
SpRL-CNN [14]	NA	NA	NA	0.460
VIEW [12]	**0.732**	0.678	0.749	0.235
Our CharRNN-BiLSTM-CRF	0.639	0.660	0.902	**0.580**

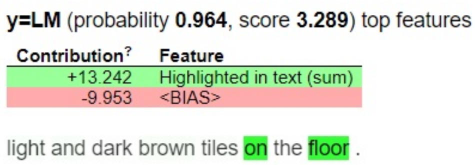


Fig. 3: LIME algorithm interpret the prediction of word "Floor" with landmark tag LM

- Exploring the use of syntactic dependencies in Word Embedding and mixing them with pre-trained ones.
- We would like to use deep contextual language embedding ELMO and Bert which achieved state-of-art performance in many NLP tasks.

References

1. Bastianelli, E., Croce, D., Nardi, D., Basili, R.: UNITOR-HMM-TK : Structured Kernel-based Learning for Spatial Role Labeling. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013). vol. 2, pp. 573–579 (2013), https://www.cs.york.ac.uk/semeval-2013/accepted/52_Paper.pdf
2. Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H.: Joint Learning of Character and Word Embeddings (Ijcai), 1236–1242 (2015)

3. Chiu, J.P., Nichols, E.: Named Entity Recognition with Bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics **4**, 357–370 (2016), http://nlp.stanford.edu/projects/glove/
4. Grübinger, M., Clough, P., Müller, H., Deselaers, T.: The IAPR TC-12 Benchmark: A New Evaluation Resource for Visual Information Systems. In: LREC Workshop OntoImage Language Resources for Content-Based Image Retrieval. pp. 13–23 (2006), http://www-i6.informatik.rwth-aachen.de/publications/download/34/The IAPR Benchmark: A New Evaluation Resource for Visual Information Systems.pdf
5. Irsoy, O., Cardie, C.: Opinion mining with deep recurrent neural networks. In: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. pp. 720–728 (2014). https://doi.org/10.3115/v1/d14-1080
6. Kolomiyets, O., Kordjamshidi, P., Moens, M.F., Bethard, S.: SemEval-2013 Task 3: Spatial Role Labeling. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) **2**(SemEval), 255–262 (2013), http://www.aclweb.org/anthology/S13-2044
7. Kordjamshidi, P., Bethard, S., Moens, M.F.: SemEval-2012 Task 3: Spatial Role Labeling. Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2012) pp. 365–373 (2012), https://dl.acm.org/citation.cfm?id=2387694
8. Kordjamshidi, P., Moens, M.F.: Global machine learning for spatial ontology population. Journal of Web Semantics **30**, 3–21 (2015). https://doi.org/10.1016/j.websem.2014.06.001, http://dx.doi.org/10.1016/j.websem.2014.06.001
9. Kordjamshidi, P., Otterlo, M.V., Moens, M.F.: From Language towards Formal Spatial Calculi (2010), https://lirias.kuleuven.be/bitstream/123456789/274901/1/paper3
10. Kordjamshidi, P., Van Otterlo, M., Moens, M.F.: Spatial role labeling. ACM Transactions on Speech and Language Processing **8**(3), 1–36 (2011). https://doi.org/10.1145/2050104.2050105, http://dl.acm.org/citation.cfm?doid=2050104.2050105
11. Kottur, S., Moura, J.M.F., Lee, S., Batra, D.: Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog (2017). https://doi.org/10.2200/S00762ED1V01Y201703HLT037, http://arxiv.org/abs/1706.08502
12. Ludwig, O., Liu, X., Kordjamshidi, P., Moens, M.F.: Deep Embedding for Spatial Role Labeling (2015), 1–19 (2016). https://doi.org/10.13140/RG.2.1.5080.4884, http://arxiv.org/abs/1603.08474
13. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers. vol. 2, pp. 1064–1074 (2016). https://doi.org/10.18653/v1/p16-1101
14. Mazalov, A.: Understanding Spatial Semantics in Natural Language (October) (2016), https://fenix.tecnico.ulisboa.pt/80462.pdf
15. Pustejovsky, J., Kordjamshidi, P., Moens, M.f., Levine, A., Dworman, S., Yocum, Z., Leuven, K.U.: SemEval-2015 Task 8: SpaceEval (1), 884–894 (2015)
16. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
17. Roberts, K., Harabagiu, S.M.: UTD-SpRL: A Joint Approach to Spatial Role Labeling pp. 419–424 (2012), http://www.aclweb.org/anthology/S12-1056