Knowledge-Based Systems 260 (2023) 110127

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Transition-based semantic role labeling with pointer networks

Daniel Fernández-González

Universidade da Coruña, CITIC, FASTPARSE Lab, LyS Group, Depto. de Ciencias de la Computación y Tecnologías de la Información, Campus de Elviña, s/n, A Coruña, 15071, Spain

A R T I C L E I N F O

Article history:
Received 18 March 2022
Received in revised form 26 October 2022
Accepted 14 November 2022
Available online 21 November 2022

Dataset link: https://github.com/danifg/SRLPointer

Keywords:
Natural language processing
Computational linguistics
Semantic role labeling
Neural network
Deep learning

A B S T R A C T

Semantic role labeling (SRL) focuses on recognizing the predicate–argument structure of a sentence and plays a critical role in many natural language processing tasks such as machine translation and question answering. Practically all available methods do not perform full SRL, since they rely on pre-identified predicates, and most of them follow a pipeline strategy, using specific models for undertaking one or several SRL subtasks. In addition, previous approaches have a strong dependence on syntactic information to achieve state-of-the-art performance, despite being syntactic trees equally hard to produce. These simplifications and requirements make the majority of SRL systems impractical for real-world applications. In this article, we propose the first transition-based SRL approach that is capable of completely processing an input sentence in a single left-to-right pass, with neither leveraging syntactic information nor resorting to additional modules. Thanks to our implementation based on Pointer Networks, full SRL can be accurately and efficiently done in $O(n^2)$, achieving the best performance to date on the majority of languages from the CoNLL-2009 shared task.
© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Semantic role labeling (SRL) has been successfully applied to a wide spectrum of *natural language processing* (NLP) applications such as *machine translation* [1–3], *information extraction* [4], *question answering* [5–7] and *text comprehension* [8], *inter alia*. This fundamental NLP task can be seen as a shallow semantic parsing that aims to extract the *"who did what to whom, how, where and when"* from an input text by identifying *predicate–argument* relations. These semantic relations are usually represented by a set of labeled dependencies, where each one connects a predicate to either the entire phrasal argument (following the *span-based* SRL formalism) or just the argument's syntactic head (following the *dependency-based* SRL annotation). While we can find recent studies that seek to improve performance on the former [9–11], this research work focuses on dependency-based SRL, which was popularized by CoNLL-2008 and CoNLL-2009 shared tasks [12,13]. An example of predicate–argument relations represented as a dependency-based SRL structure is depicted in Fig. 1(a).

SRL is traditionally decomposed into four simpler subtasks: *predicate identification* (*e.g., managers* in Fig. 1(a)), *predicate sense disambiguation* (*manager.01* is the sense of predicate *managers* in the example), *argument identification* (*e.g., fund*) and *argument role labeling* (*fund* is argument A1 for predicate *managers*). While the four subtasks had to be completed in the CoNLL-2008 shared task, CoNLL-2009 corpora notably simplified SRL by providing pre-identified predicates beforehand. This simplification, coupled with the fact that the vast majority of approaches are only tested on the CoNLL-2009 benchmark, resulted in the common practice of not performing full SRL and exclusively focusing on the last three subtasks.

Furthermore, we can find in the literature as most SRL systems adopt either a *pipeline* framework [14–17] or an *end-to-end* strategy [18–20]. While the former approach resorts to different specific models to separately address one or two subtasks, the latter strategy employs a single model to accomplish the SRL task. However, it is worth mentioning that the vast majority of these end-to-end approaches do not perform predicate identification and, therefore, are not considered full SRL systems.

Moreover, most previous efforts mainly focused on *syntax-aware* SRL methods [16–18,21,22]: *i.e.*, they leverage syntactic information (also provided by the CoNLL-2009 corpora) to produce state-of-the-art accuracies. These approaches were motivated, among other reasons, by the fact that a significant portion of arcs from syntactic dependency trees matches predicate–argument relations in SRL structures. Nevertheless, it is important to note that dependency parsing is also an equally challenging and resource-consuming NLP task, and syntactic training data can be especially scarce in low-resource languages.

The fact that practically all approaches heavily rely on information that is not always available (such as gold predicates and syntactic dependency trees) makes them impractical for real-life downstream applications that require full SRL. In addition,

E-mail address: d.fgonzalez@udc.es.
URL: https://danifg.github.io.

https://doi.org/10.1016/j.knosys.2022.110127
0950-7051/© 2022 Elsevier B.V. All rights reserved.

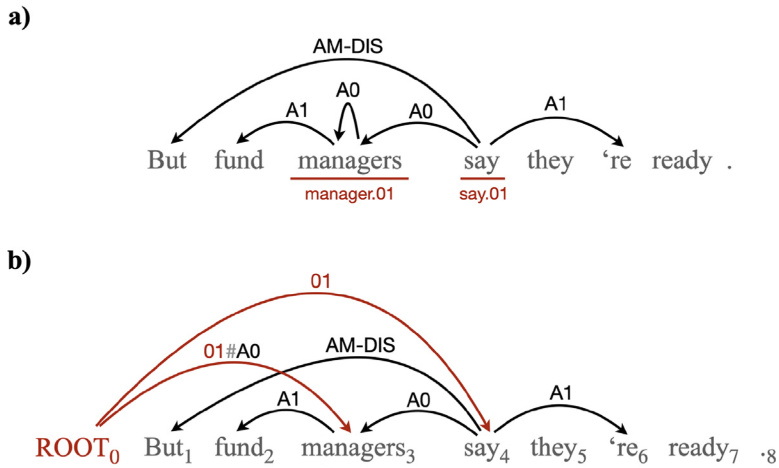

Fig. 1. **(a)** Example of a dependency-based SRL structure from the CoNLL-2009 English data and **(b)** its labeled dependency graph representation for end-to-end modeling.

the use of a single model in end-to-end architectures not only mitigates the error-propagation problem in pipeline strategies, but also notably simplifies the decoding process.

To alleviate these inconveniences, Cai et al. [23] introduced the first end-to-end *syntax-agnostic* approach for accurately performing full SRL. They handle the four subtasks by a single *graph-based* model, following a two-stage decoding procedure: they first identify and classify all predicates and then search for arguments and semantic roles for each of these predicates. The latter is implemented by independently scoring all possible predicate–argument dependencies and then exhaustively searching for a high-scoring graph by combining these scores. This work was recently improved by Zhou et al. [24]. They proposed a syntax-agnostic method that jointly identifies and classifies predicates and arguments in a single stage. In addition, their approach leverages high-order information, scoring sets of predicate–argument dependencies and computing the high-scoring graph in cubic time. The resulting graph-based model achieves a remarkable performance in full end-to-end SRL.

Unlike graph-based methods, *transition-based* algorithms were barely proposed for SRL modeling. These generate a sequence of actions (transitions) to incrementally build predicate–argument dependencies (usually from left to right). This is typically done by local, greedy prediction and can efficiently process a sentence in a linear or quadratic number of actions. Although they provide higher efficiency than graph-based models and were successfully employed in other parsing tasks [25–28], only two transition-based SRL systems were presented [29,30], neither of them performing full syntax-agnostic SRL.

In this article, we propose the first transition-based approach for full syntax-agnostic SRL. Our model does not rely on any kind of syntactic information, even discarding part-of-speech (PoS) tags,[1] and incrementally produces predicate–argument dependencies in a single left-to-right pass. For implementing our technique, we resort to Pointer Networks [31], which provide an efficient $O(n^2)$ runtime complexity in practice. We experimentally prove that our end-to-end SRL system surpasses strong baselines (including syntax-aware approaches) on the CoNLL-2009 corpora, becoming the highest-performing model in practically all languages. Our major contributions can be summarized as follows:

- We design the first syntax-agnostic approach for transition-based SRL, requiring neither syntactic dependency trees nor PoS tag information.
- Our approach is end-to-end and can be directly applied to plain text, tackling the four SRL subtasks in one shot and without requiring any external module.
- Our model is robust and achieves state-of-the-art results on the majority of languages from CoNLL-2009 benchmark with and without pre-identified predicates.
- We empirically prove that the proposed transition-based technique processes CoNLL-2009 corpora in $O(n^2)$ time, being more efficient than best-performing graph-based models ($O(n^3)$).

The remainder of this article is organized as follows: Section 2 introduces previous studies for syntax-agnostic SRL. In Section 3, we define the transition system for full SRL and detail the proposed Pointer Network architecture. In Section 4, we extensively evaluate our SRL model on CoNLL-2009 corpora, present a discussion of the experimental results, analyze the contribution of each component and study the time complexity of our approach in practice. Lastly, Section 5 includes final conclusions.

2. Related work

Syntax-based approaches [14,16–18,22] have been the mainstream for dependency-based SRL, consistently proving that syntactic information is highly effective for achieving state-of-the-art accuracies.

Alternatively, Marcheggiani et al. [15] proposed the first syntax-agnostic model for dependency-based SRL. Instead of leveraging syntactic features for capturing long-distance predicate–argument dependencies, they employ a BiLSTM-based encoder. Their approach exclusively addresses argument identification and labeling, directly using gold predicates from CoNLL-2009 corpora and resorting to other works [14,32,33] for language-specific predicate disambiguation. This initial attempt was followed by the graph-based model by Cai et al. [23], which is considered the first syntax-agnostic method for full end-to-end SRL. They perform the four SRL subtasks by a single model. To achieve that, they apply the widely-used *biaffine* attention mechanism [34] for exhaustively scoring all predicate–argument relations and their semantic roles; then, during decoding, they first search for predicates and, in a second stage, each predicate is processed by identifying and labeling its arguments. This work was improved by other syntax-agnostic graph-based techniques such as [35]

[1] PoS tags are considered lexical-level syntactic features and, therefore, a truly syntax-agnostic approach should not leverage that information.

and [19] that, instead of applying a two-stage strategy, model dependency-based SRL as a graph parsing task, where predicates and arguments are uniformly treated and jointly processed. However, while performing full SRL, these two systems follow a pipeline strategy and, therefore, are not considered end-to-end: the former follows [14] for predicate disambiguation and the latter identifies predicates in advance with a separate sequence tagging model. Recently, Zhou et al. [24] extended the work by Li et al. [19] to full end-to-end SRL, obtaining promising results on the CoNLL-2009 English dataset without pre-identified predicates.

We can also find recent syntax-agnostic approaches that do not perform full SRL and follow a predicate-centered strategy for decoding and word representation (based on gold predicates provided by CoNLL-2009 corpora): [36] and [37], which additionally implement different iterative refinement procedures, and [20], which employs an additional specific encoder for contextualizing each gold predicate in the sentence.

Regarding transition-based SRL modeling, just two syntax-aware attempts were proposed. Choi and Palmer [29] presented a pre-deep-learning system that relies on handcrafted syntactic features and is only able to identify arguments and predict their semantic roles. And, recently, Fei et al. [30] developed an end-to-end system that resorts to TreeLSTMs [38] for leveraging syntactic information. In addition, this work firstly processes the sentence from left to right in order to identify any possible predicate and, when a predicate is found, their transition system searches for arguments from near to far. Finally, while their approach can be used for full SRL, Fei et al. [30] did not evaluate their transition-based model on CoNLL-2009 corpora without gold predicates, just testing it on the English dataset with pre-identified predicates.

On the other hand, graph-based approaches [39,40] are also the mainstream in other semantic parsing tasks such as Semantic Dependency Parsing (SDP) [41]; however, Fernández-González and Gómez-Rodríguez [27] introduced a transition-based algorithm that yields state-of-the-art scores on that task. Inspired by the latter, we design the first transition-based model that, without any kind of syntactic features, performs full end-to-end SRL in a single forward pass.

3. Model

3.1. Graph representation

For full end-to-end SRL modeling, the four subtasks must be formulated as a single graph parsing task, where predicates and arguments are uniformly treated. To that end, the relations between predicates and arguments must be represented as labeled dependency arcs, where the predicate acts as the semantic *head*, the argument as the semantic *dependent* and the semantic role as the *dependency label*. In addition, end-to-end graph-based SRL models [16,23,35] augment the original dependency-based structure by building a single-rooted graph (not necessarily acyclic): *i.e.*, all predicates are attached to an artificial root node (added at the begging of the sentence) and the resulting arcs are labeled with predicate sense tags.[2] Formally, given an input sentence $X = w_0, w_1, \dots, w_n$ (being w_0 the artificial root node), a full end-to-end SRL system is expected to completely produce a graph G represented as a set of labeled predicate–argument relations: $G \subseteq W \times W \times L$, where W is the set of input words ($W = \{w_0, w_1, \dots, w_n\}$) and L refers to the set of semantic role plus predicate sense labels. We adopt this graph representation for implementing our transition-based model. In Fig. 1(b), we present the resulting single-rooted graph obtained from the original dependency-based SRL structure in Fig. 1(a).

Moreover, it is worth noting that, in some languages such as Czech, the same word can serve as two or more arguments to the same predicate, resulting in two or more dependencies between the same two words. For instance, we can find two dependency arcs between predicate w_p and word w_a respectively tagged with semantic roles A1 and A2 (meaning that w_a serves as arguments A1 and A2 for predicate w_p). We handle this by keeping just one dependency between words w_p and w_a, and by assigning the concatenation of both semantic roles (A1|A2) as dependency label. Additionally, in some datasets, a predicate can be also an argument of itself (*i.e.*, a dependency where the head and the dependent are the same word). An example of this can be seen in Fig. 1, where the word *managers* acts as the predicate and argument A0. To encode that information in our final representation, we concatenate the semantic role label of this dependency (A0 in the example) with the predicate sense (01 for predicate *managers*) and use the resulting label (01#A0) for tagging the arc that will attach that predicate to the artificial root. In both pre-processing strategies, the original structure is easily recovered before evaluation.

3.2. Transition system

Inspired by [27], we design a transition system for generating a graph G for the input sentence by applying a sequence of actions $A = a_1, \dots, a_t$. These actions (or *transitions*) will be sequentially predicted by a neural model. In this section, we formally define the main components of the proposed transition system: *state configurations* and *actions*.

While other transition-based SRL systems require more complex state configurations with several stacks and additional data structures for temporarily storing partially-processed words [29,30], we just need to implement two pointers for building any dependency graph. More in detail, the proposed transition system has state configurations of the form $c = \langle i, j, \Sigma \rangle$, where i points at the word w_i currently being processed, j indicates the position of the last identified predicate w_j for w_i and Σ contains the set of already-created edges. Given a sentence $X = w_0, w_1, \dots, w_n$ (with w_0 as artificial root node), the process starts at the initial state configuration $c_{initial} = \langle 1, -1, \emptyset \rangle$, where i is pointing at the first input word w_1, no predicate position has been saved yet at j and Σ is empty. Then, after applying a sequence of actions A, the transition system reaches a final configuration of the form $c_{terminal} = \langle n+1, -1, \Sigma \rangle$, where all the words have been shifted and Σ contains the edges of the graph G for the input sentence X.

Unlike the works by Choi and Palmer [29] and Fei et al. [30] that define six different actions to produce SRL structures, we just require two transitions:

- Arc-p that attaches the current focus word w_i to the head word at position p, building a semantic dependency arc from the identified predicate w_p to argument w_i. By applying this action, the transition system moves from state configurations $\langle i, j, \Sigma \rangle$ to $\langle i, p, \Sigma \cup \{w_p \to w_i\} \rangle$. This transition can only be applied if the resulting edge has not been created yet (*i.e.*, $w_p \to w_i \notin \Sigma$) and the predicate w_p is in a higher position than the last identified predicate for w_i in position j (*i.e.*, $j < p$). The latter condition is necessary since the head assignment to the current focus word must follow the

[2] As common practice, the lemma is removed from the predicate sense tag (*e.g.*, *manager* in *manager.01*), since that information is typically provided in datasets from CoNLL-2009.

Table 1
Transition sequence and resulting state configurations for incrementally generating arcs of the dependency graph in Fig. 1(b).

Transition	State configuration	Focus word$_i$	Last predicate$_j$	Added arc
	$\langle 1, -1, \Sigma \rangle$	But$_1$		
Arc-4	$\langle 1, 4, \Sigma \cup \{4 \rightarrow 1\} \rangle$	But$_1$	say$_4$	say$_4$ $\rightarrow$ But$_1$
Shift	$\langle 2, -1, \Sigma \rangle$	fund$_2$		
Arc-3	$\langle 2, 3, \Sigma \cup \{3 \rightarrow 2\} \rangle$	fund$_2$	managers$_3$	managers$_3$ $\rightarrow$ fund$_2$
Shift	$\langle 3, -1, \Sigma \rangle$	managers$_3$		
Arc-0	$\langle 3, 0, \Sigma \cup \{0 \rightarrow 3\} \rangle$	managers$_3$	Root$_0$	Root$_0$ $\rightarrow$ managers$_3$
Arc-4	$\langle 3, 4, \Sigma \cup \{4 \rightarrow 3\} \rangle$	managers$_3$	say$_4$	say$_4$ $\rightarrow$ managers$_3$
Shift	$\langle 4, -1, \Sigma \rangle$	say$_4$		
Arc-0	$\langle 4, 0, \Sigma \cup \{0 \rightarrow 4\} \rangle$	say$_4$	Root$_0$	Root$_0$ $\rightarrow$ say$_4$
Shift	$\langle 5, -1, \Sigma \rangle$	they$_5$		
Shift	$\langle 6, -1, \Sigma \rangle$	're$_6$		
Arc-4	$\langle 6, 4, \Sigma \cup \{4 \rightarrow 6\} \rangle$	're$_6$	say$_4$	say$_4$ $\rightarrow$ 're$_6$
Shift	$\langle 7, -1, \Sigma \rangle$	ready$_7$		
Shift	$\langle 8, -1, \Sigma \rangle$	.$_8$		
Shift	$\langle 9, -1, \Sigma \rangle$			

left-to-right order used for training.[3] Finally, the resulting dependency arc is labeled by a jointly-trained classifier, as described in the following section.

- Shift that moves i one position to the right, pointing at the word w_{i+1}, and, since we will start searching for predicates for that unprocessed word, j is initialized to -1. Therefore, we move from state configurations $\langle i, j, \Sigma \rangle$ to $\langle i+1, -1, \Sigma \rangle$ by using this action.

The resulting transition system processes an input sentence from left to right by applying a sequence of Shift-Arc actions that attaches some words to one or several predicates and leaves others unattached, incrementally building a dependency graph. Please see in Table 1 how this transition-based algorithm generates the graph in Fig. 1(b).

While the described transition system was originally designed for performing full end-to-end SRL, it can be easily adapted to leverage gold predicate information provided by the CoNLL-2009 corpora (where no predicate identification is required). For that purpose, a third condition must be added to the Arc-p transition: it can be applied only if the word w_p is a gold predicate. In addition, each word w_i that is a gold predicate is directly attached to the artificial root w_0. Please note that, while these modifications allow a fairer comparison on the CoNLL-2009 benchmark, our approach does not follow a predicate-centered strategy as the vast majority of SRL systems evaluated on that shared task. These typically base their training and decoding procedures on the existence of pre-identified predicates and, as a consequence, they simply focus on individually processing each given predicate by searching for its arguments over the whole input.

3.3. Pointer Network

We use a Pointer Network [31] for implementing the proposed transition-based algorithm. We manage to properly represent state configurations in this neural model and use that information for generating transition sequences necessary for processing input sentences. We detail below the different components of our neural architecture:

Word representation. Each input token w_i is represented by the concatenation of word (e_i^{word}), lemma (e_i^{lemma}) and character-level (e_i^{char}) embeddings. The latter is obtained by encoding characters inside w_i with convolutional neural networks (CNNs) [42] and, unlike most SRL systems, we do not include PoS tag embeddings in order to develop a truly syntax-agnostic approach.

In addition, we also evaluate our model with deep contextualized word embeddings (e_i^{BERT}) from pre-trained language model BERT [43]. In particular, we use mean pooling (*i.e.*, the average value of all subword embeddings) to extract word-level representations from weights of the second-to-last layer. Lastly, we follow previous works [16,19,23] and leverage a predicate indicator embedding ($e_i^{indicator}$)[4] when our model is tested following the CoNLL-2009 setting and, therefore, pre-identified predicates are available. When all these embeddings are exploited, the word representation e_i is obtained as follows:

$$e_i = e_i^{word} \oplus e_i^{lemma} \oplus e_i^{char} \oplus e_i^{BERT} \oplus e_i^{indicator}$$

Encoder. As a common practice, we feed a three-layer bidirectional LSTM (BiLSTM) [44] to generate a context-aware vector representation h_i for each word vector e_i:

$$h_i = \text{BiLSTM}(e_i) = f_i \oplus b_i$$

where f_i and b_i are respectively the forward and backward hidden states of the last LSTM layer at the ith position. Additionally, a randomly-initialized vector h_0 is used for denoting the artificial root node. As a result, the sequence of word representations $E = e_1, \ldots, e_n$ is encoded into a sequence of encoder vectors $H = h_0, h_1, \ldots, h_n$.

Decoder. We employ a unidirectional one-layer LSTM plus an attention mechanism for decoding. Firstly, at each time step t, state configurations $c_t = \langle i, j, \Sigma \rangle$ are encoded by feeding the LSTM with the combination[5] of the respective encoder representations h_i and h_j of the current focus word (w_i) and its last assigned predicate (w_j) if available. This will generate the state configuration representation s_t[6]:

$$r_t = h_i + h_j$$

$$s_t = \text{LSTM}(r_t)$$

Please note that, while introducing high-order information (such as the co-parent representation h_j of a future predicate of w_i) significantly penalizes graph-based models' runtime complexity, transition-based algorithms can straightforwardly leverage this information without harming their efficiency.

[3] Please note that, while applying a different order in head attachments (*e.g.*, inside-out [25]) could lead to slight improvements in accuracy, we decided to keep it simple and follow a left-to-right strategy.

[4] This embedding simply marks whether w_i is a predicate or not based on the information provided by CoNLL-2009 corpora.

[5] Instead of concatenating both vectors, we compute the element-wise summation to avoid increasing the dimension of the resulting vector r_t.

[6] Please note that Σ in c_t is not used for generating the state configuration representation s_t. Σ was exclusively designed to collect already-built edges and its main purpose is to prevent the transition system from creating dependency arcs already added to Σ.

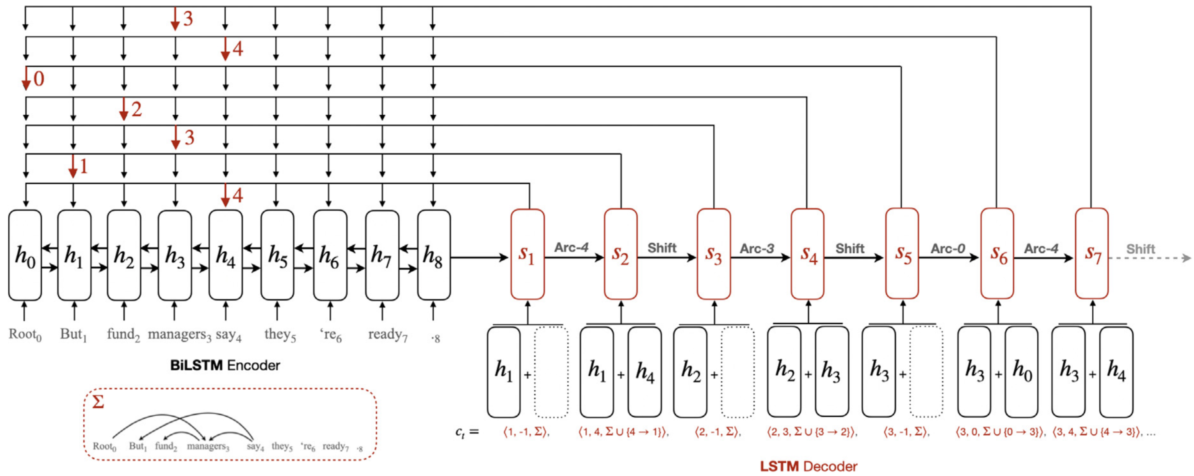

Fig. 2. Pointer Network architecture and decoding procedure for partially generating the dependency graph in Fig. 1(b).

Once the current state configuration c_t is properly represented as s_t, an attention mechanism is used for selecting the action a_t to be applied at time step t. In particular, this mechanism employs the biaffine scoring function [34] to compute the score between each input word w_k (represented by the encoder vector h_k with $k \in [0, n]$) and the state configuration representation s_t; and then normalizes the resulting vector v^t of length n to output the attention vector α_t, which is a *softmax* distribution with dictionary size equal to the length of the input:

$$v_k^t = \text{score}(s_t, h_k) = f_1(s_t)^T W f_2(h_k) \quad + U^T f_1(s_t) + V^T f_2(h_k) + b$$

$$\alpha_t = \text{softmax}(v^t)$$

where W is the weight matrix of the bi-linear term, U and V are the weight tensors of the linear terms, b is the bias vector and $f_1(\cdot)$ and $f_2(\cdot)$ are two one-layer perceptrons with ELU activation to obtain lower-dimensionality and avoid overfitting. From the attention vector α_t, we select the highest-scoring position p_t from the input and, being $c_t = \langle i, j, \Sigma \rangle$, use that information to choose the current action a_t between the two available transitions as follows:

- if $p_t = i$, then a Shift action is applied, moving the focus word pointer to the next word.
- On the contrary, if $p_t \neq i$, then the Arc transition parameterized with p_t will be considered, building a dependency arc between the word w_i and its predicate w_{p_t}. In case that conditions required by this action are not satisfied, then the next highest-scoring position in α_t will be used for choosing again between the Shift and Arc transitions.

In Fig. 2, we include a sketch of the proposed Pointer Network architecture and decoding steps for partially building the graph structure in Fig. 1(b).

Finally, when an Arc-p_t transition creates an edge between the current focus word w_i and the predicate w_{p_t}, we apply a classifier for labeling it. This labeler is implemented by the biaffine scoring function previously described. Concretely, for each available semantic role or predicate sense label $l \in L$, we compute the score of assigning l to the predicted arc between the argument w_i (encoded as s_t) and the predicate w_{p_t} (represented by h_{p_t}) as follows:

$$u_l^t = \text{score}(s_t, h_{p_t}, l) = g_1(s_t)^T W_l g_2(h_{p_t}) + U_l^T g_1(s_t) + V_l^T g_2(h_{p_t}) + b_l$$

$$\beta_t = \text{softmax}(u^t)$$

where W_l is a weight matrix, U_l and V_l are weight tensors and b_l is the bias vector exclusively used for each label l, and $g_1(\cdot)$ and $g_2(\cdot)$ are two one-layer perceptrons with ELU activation. Then, we select the highest-scoring label in the vector β_t for tagging the predicted arc.

Training objectives. The loss of our model comes from the transition system and the labeler. On the one hand, the transition system is trained by minimizing the total log loss of choosing the correct sequence of Shift-Arc transitions A to output the gold dependency graph G for the input sentence X (*i.e.*, predicting the correct sequence of indices p_t, with each decision at time step t being conditioned by previous ones ($p_{<t}$)):

$$\mathcal{L}_{tran}(\theta) = -\sum_{t=1}^{T} log P_\theta(p_t | p_{<t}, X)$$

On the other hand, the labeler is trained to minimize the total log loss of assigning the correct label l, given a dependency arc from predicate w_{p_t} to dependent word w_i:

$$\mathcal{L}_{label}(\theta) = -\sum_{t=1}^{T} log P_\theta(l | w_{p_t}, w_i)$$

Finally, we jointly train the transition system and the labeler by optimizing the sum of their losses:

$$\mathcal{L}(\theta) = \mathcal{L}_{tran}(\theta) + \mathcal{L}_{label}(\theta)$$

4. Experiments

4.1. Data

We conduct experiments on datasets from CoNLL-2008 and CoNLL-2009 shared tasks [12,13]. The former was an English-only benchmark with in-domain (from the Wall Street Journal corpus [45]) and out-of-domain (from the Brown Corpus [46]) test sets. This was extended by the CoNLL-2009 shared task to a multilingual benchmark by adding 6 more languages (Catalan, Chinese, Czech, German, Japanese and Spanish) with 6 in-domain and 2 out-of-domain test sets. Unlike CoNLL-2008 English dataset, CoNLL-2009 corpora notably simplified the SRL task by providing gold predicates beforehand.[7] In order to properly test our model under a real-world usage scenario and also compare it

[7] Please note that the CoNLL-2009 shared task also augmented CoNLL-2008 English dataset with pre-identified predicates.

Table 2
Data statistics for CoNLL-2009 training sets. We report the number of sentences, annotated sentences (with at least one predicate), tokens, predicates and arguments, as well as the average sentence length.

Language	Sentences	Annotated	Avg. length	Tokens	Predicates	Arguments
Catalan (CA)	13,200	12,873	30.2	390,302	37,431	84,367
Chinese (ZH)	22,277	21,071	28.5	609,060	102,813	231,869
Czech (CZ)	38,727	38,578	16.9	652,544	414,237	365,255
English (EN)	39,279	37,847	25.0	958,167	179,014	393,699
German (DE)	36,020	14,282	22.2	648,677	17,400	34,276
Spanish (ES)	14,329	13,835	30.7	427,442	43,824	99,054

to most previous works, we evaluate our approach on CoNLL-2009 datasets[8] following two different settings: *w/ pre-identified predicates* and *w/o pre-identified predicates* (requiring the latter full SRL as in the CoNLL-2008 shared task).

In Table 2, we summarize the training data statistics for each language. From this information, we can see as, while having a notable amount of sentences in the training set, German is considered a low-resource language due to the low proportion of annotated sentences and predicate instances in comparison to other languages [20]. For the same reasons, Catalan and Spanish are also classified as low-resource languages.

Finally, we use the CoNLL-2009 official scoring script[9] for performance evaluation. This measures the labeled precision, recall and F_1 score for semantic dependencies.

4.2. Setup

In our experiments, word and lemma embeddings are initialized with 300-dimensional GloVe vectors [47] for English; structured-skipgram embeddings [48] for Chinese (dimension 80), German (dimension 64) and Spanish (dimension 64); and 64-dimensional Polyglot embeddings for Catalan and Czech. 100-dimensional character-level embeddings are randomly initialized and, for CNNs, we use 100 filters with a window size of 3 and max-pooling. For BERT-based embeddings, we extract respectively 768-dimensional and 1024-dimensional vectors from specific $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$ models [43]: `bert-large-cased` for English, `bert-base-multilingual-cased` for Catalan, `bert-base-chinese` for Chinese, `deepset/gbert-large` [49] for German, `bert-base-bg-cs-pl-ru-cased` [50] for Czech and `bert-base-spanish-wwm-cased` [51] for Spanish. Following a greener and less resource-consuming strategy, BERT-based embeddings are not fine-tuned during training. Finally, we randomly initialize a 16-dimensional predicate indicator embedding under the *w/ pre-identified predicate* setup.

Most hyperparameters were taken from [27] and we directly apply them to all datasets and languages without further optimization. For training, we employ Adam optimizer [52] with initial learning rate of $\eta_0 = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.9$. We also use a fixed decay rate of 0.75 and a gradient clipping of 5.0 in order to mitigate the gradient exploding effect [53]. Moreover, we use LSTMs with 512-dimensional hidden states for both encoder and decoder, applying recurrent dropout [54] with a drop rate of 0.33 between hidden states and layers. We also apply a 0.33 dropout to all embeddings. In addition, all models are trained up to 600 epochs with batch size 32, and beam-search decoding with beam size 5 is utilized in all experiments. Finally, we choose the checkpoint with the highest labeled F_1 score on the development set for posterior in-domain and out-of-domain evaluations.

4.3. Results and discussion

English results. In Table 3, we compare our model against previous full SRL systems on English in-domain and out-of-domain tests sets under the *w/o pre-identified predicates* setting. Our single-model approach achieves the best performance on both test sets among syntax-agnostic SRL systems without deep contextualized word embeddings, even outperforming all syntax-based models on the WSJ test set. When pre-trained language models come into play, our system obtains competitive accuracies (improving over again all syntax-aware approaches on the in-domain test set), but it is slightly surpassed by those graph-based models [19,24] that adapt BERT-based embeddings to SRL by fine-tuning them during training.

Table 4 presents the results on English in-domain and out-of-domain tests sets *w/ pre-identified predicates*. It can be seen as predicate-centered SRL systems outnumber those developed for full SRL (which are reported in Table 3). While our approach was originally designed for dealing with the lack of gold predicates and just minor adaptations were undertaken for leveraging pre-identified predicate information, our system behaves similarly to the full SRL setting: it is the best-performing syntax-agnostic approach without contextualized word representations (also surpassing syntax-aware systems on the WSJ test set), but the highest scores are reported by graph-based models that fine-tune BERT-based embeddings [19] or, while keeping them frozen, leverage syntactic information [22].

Multilingual results. Table 5 summarizes the performance on the remaining CoNLL-2009 languages (including out-of-domain test sets if available) under both *w/ pre-identified predicates* and *w/o pre-identified predicates* settings. Compared with previous methods, our approach yields strong performance consistent across languages, regardless of the availability of gold predicate information.

Without contextualized word representations, our end-to-end SRL system improves over the best syntax-agnostic methods on all languages in the in-domain setting (with and without pre-identified predicates), bringing notable improvements on both high-resource (*e.g.*, Czech) and low-resource (*e.g.*, German) datasets. Our proposal also outperforms syntax-based models in all datasets except German, meaning that leveraging syntactic information [62] is still crucial for obtaining state-of-the-art results on that language.

When our model is augmented with frozen BERT-based embeddings, we achieve the highest score to date on 3 out of 5 languages in the in-domain setting (with and without given predicates), being only outperformed by [19] on Catalan and Spanish, where fine-tuning contextualized word vectors yields substantial accuracy gains in these low-resource datasets.

Regarding results on the out-of-domain data (with and without gold predicates), our model (with and without BERT-based embeddings) outperforms existing SRL systems (including syntax-aware approaches) by a wide margin on Czech and German test sets (being especially challenging the latter, since it contains numerous infrequent predicates specifically included for the CoNLL-2009 shared task). Lastly, we observe in Czech as adding deep

[8] We do not evaluate our approach on the Japanese dataset since it is no longer available due to licensing problems.
[9] https://ufal.mff.cuni.cz/conll2009-st/scorer.html

Table 3
Precision (P), recall (R) and F_1 scores obtained by full SRL systems on the CoNLL-2008/CoNLL-2009 English in-domain (Wall Street Journal, WSJ) and out-of-domain (Brown) test sets *w/o pre-identified predicates*. The first block gathers methods enhanced with syntactic information and, the second block, those that are syntax-agnostic. We also indicate in column "*end-to-end*" whether approaches follow an end-to-end (y) or a pipeline (n) strategy, using the latter one or more extra models to accomplish full SRL. *+ELMO* and *+BERT* stand for augmentations with deep contextualized word-level embeddings from pre-trained language models ELMO [55] and BERT, respectively; and *+Joint* means that the system learns SRL jointly with other tasks. Please note that we keep BERT-based embeddings frozen in order to avoid increasing the computational cost, and we denote with "*fine-tuned*" those systems that do undertake an expensive fine-tuning during training in order to adapt them to SRL. Finally, we mark with † those truly syntax-agnostic models that do not leverage PoS tag information.

System	End-to-end	WSJ			Brown		
		P	R	F_1	P	R	F_1
(Syntax-aware)							
He et al. [16]	n	83.9	82.7	83.3	–	–	–
Zhou et al. [56] + Joint	n	**84.2**	**87.5**	**85.9**	**76.5**	**78.5**	**77.5**
Zhou et al. [56] + Joint + BERT$_{\text{fine-tuned}}$	n	**87.4**	**89.0**	**88.2**	**80.3**	**82.9**	**81.6**
Munir et al. [57] + ELMO	n	85.8	84.4	85.1	74.6	74.8	74.7
Li et al. [22] + ELMO	n	86.2	86.0	86.1	73.8	74.6	74.2
(Syntax-agnostic)							
Cai et al. [23]	y	84.7	85.2	85.0	–	–	72.5
Li et al. [35]†	n	–	–	85.1	–	–	–
Li et al. [19]†	n	86.0	85.6	85.8	74.4	73.3	73.8
Zhou et al. [24]†	y	**86.7**	86.2	86.5	**75.8**	74.6	75.2
This work†	y	85.9	**88.0**	**86.9**	74.4	**76.4**	**75.4**
Li et al. [35]† + ELMO	n	84.5	86.1	85.3	74.6	73.8	74.2
Li et al. [19]† + BERT$_{\text{fine-tuned}}$	n	**88.6**	88.6	88.6	**79.9**	79.9	79.9
Zhou et al. [24]† + BERT$_{\text{fine-tuned}}$	y	87.6	**90.2**	**88.9**	79.0	**82.2**	**80.6**
This work† + BERT	y	87.2	89.8	88.5	**79.9**	81.8	80.4

Table 4
Precision, recall and F_1 scores achieved by state-of-the-art SRL systems on the CoNLL-2009 English in-domain (WSJ) and out-of-domain (Brown) test sets *w/ pre-identified predicates*. We use the same abbreviations described in Table 3 and, additionally, *+Semi* indicates semi-supervised training and *+Ens* specifies ensemble system. Please note that, under the *w/ pre-identified predicates* setting, a single-model system only has to perform the three remaining SRL subtasks to be considered an end-to-end approach. Finally, we do not include the syntax-aware transition-based model by Fei et al. [30] since they do not apply the CoNLL-2009 official scoring script and separately report the accuracy on predicate disambiguation and argument identification+labeling.

System	End-to-end	WSJ			Brown		
		P	R	F_1	P	R	F_1
(Syntax-aware)							
Lei et al. [58]	n	–	–	86.6	–	–	75.6
FitzGerald et al. [59] + Ens	n	–	–	86.7	–	–	75.2
Roth and Lapata [14] + Ens	n	90.3	85.7	87.9	79.7	73.6	76.5
Marcheggiani and Titov [60] + Ens	n	**90.5**	87.7	89.1	80.8	77.1	78.9
He et al. [16]	y	89.7	89.3	89.5	81.9	76.9	79.3
Cai and Lapata [17] + Joint	n	**90.5**	88.6	89.6	80.5	78.2	79.4
Kasai et al. [21]	n	89.0	88.2	88.6	78.0	77.2	77.6
He et al. [18]	n	90.0	**90.0**	**90.0**	–	–	–
Zhou et al. [56] + Joint	n	88.7	89.8	89.3	**82.5**	**83.2**	**82.8**
Li et al. [22]	n	–	–	89.2	–	–	80.1
Li et al. [61] + ELMO	n	90.3	89.3	89.8	80.6	79.0	79.8
Cai and Lapata [17] + Joint + ELMO	n	90.9	89.1	90.0	80.8	78.6	79.7
Cai and Lapata [62] + ELMO	n	91.1	90.4	90.7	82.1	81.3	81.6
Cai and Lapata [62] + ELMO + Semi	n	**91.7**	90.8	91.2	83.2	81.9	82.5
Kasai et al. [21] + ELMO	n	90.3	90.0	90.2	81.0	80.5	80.8
He et al. [18] + BERT	n	90.4	91.3	90.9	–	–	–
Zhou et al. [56] + Joint + BERT$_{\text{fine-tuned}}$	n	91.2	91.2	91.2	**85.7**	**86.1**	**85.9**
Munir et al. [57] + ELMO	n	91.2	90.6	90.9	83.1	82.6	82.8
Li et al. [22] + ELMO	n	90.5	**91.7**	91.1	83.3	80.9	82.1
Li et al. [22] + BERT	n	–	–	**91.8**	–	–	83.2
(Syntax-agnostic)							
Marcheggiani et al. [15]	n	88.7	86.8	87.7	79.4	76.2	77.7
Cai et al. [23]	y	89.9	89.2	89.6	79.8	78.3	79.0
Li et al. [19]†	y	**91.3**	88.7	90.0	**81.8**	78.4	80.0
This work†	y	90.2	**90.5**	**90.4**	80.6	**80.5**	**80.6**
Li et al. [35]† + ELMO	n	89.6	91.2	90.4	81.7	81.4	81.5
Chen et al. [36]† + ELMO	y	90.7	91.4	91.1	82.7	82.8	82.7
Lyu et al. [37] + ELMO	n	–	–	91.0	–	–	82.2
Conia and Navigli [20]† + BERT	y	91.2	**91.8**	91.5	–	–	84.6
Li et al. [19]† + BERT$_{\text{fine-tuned}}$	y	**92.6**	91.0	**91.8**	**86.5**	83.8	**85.1**
This work† + BERT	y	91.3	91.6	91.4	84.5	**84.2**	84.4

Table 5
F_1 scores on the remaining CoNLL-2009 in-domain (*id*) and out-of-domain (*ood*) test sets for both *w/ pre-identified predicates* (top) and *w/o pre-identified predicates* (bottom) settings. *CoNLL-2009 ST best* refers to the best F_1 scores reported for the CoNLL-2009 shared task [13]. We use the same abbreviations previously described in Tables 3 and 4.

System (*w/ pre-identified predicates*)	**End-to-end**	**CA**$_{id}$	**CZ**$_{id}$	**CZ**$_{ood}$	**DE**$_{id}$	**DE**$_{ood}$	**ES**$_{id}$	**ZH**$_{id}$
(Syntax-aware)								
CoNLL-2009 ST best		80.3	86.5	**85.4**	79.7	65.9	80.5	78.6
Zhao et al. [32]	n	80.3	85.2	82.7	76.0	**67.8**	80.5	77.7
Roth and Lapata [14] + Ens	n	–	–	–	80.1	–	80.2	79.4
Cai and Lapata [17] + Joint	n	–	–	–	82.7	–	81.8	83.6
Cai and Lapata [62]	n	–	–	–	83.3	–	82.1	84.6
Cai and Lapata [62] + Semi	n	–	–	–	**83.8**	–	82.9	**85.0**
He et al. [18]	y	**84.9**	**88.8**	–	78.5	–	**83.9**	84.8
He et al. [18] + BERT	y	**86.0**	89.7	–	**81.1**	–	**85.2**	**86.9**
Li et al. [22] + ELMO	n	85.5	**90.5**	–	76.6	–	84.3	86.1
(Syntax-agnostic)								
Marcheggiani et al. [15]	n	–	86.0	87.2	–	–	80.3	81.2
Mulcaire et al. [63]†	y	79.5	85.1	–	70.0	–	77.3	81.9
Chen et al. [36]	y	81.7	88.1	–	76.4	–	81.3	81.7
Lyu et al. [37]	n	80.9	87.6	86.0	75.9	65.7	80.5	83.3
Li et al. [19]†	y	84.7	90.6	90.8	76.4	70.1	84.2	86.0
This work†	y	**85.9**	**93.3**	**92.5**	**79.5**	**71.0**	**85.2**	**86.7**
Conia and Navigli [20]† + BERT	y	86.2	90.1	90.6	86.5	73.1	85.3	87.3
Li et al. [19]† + $\text{BERT}_{\text{fine-tuned}}$	y	**86.8**	91.6	91.7	85.5	71.9	**86.9**	88.5
This work† + BERT	y	86.2	**93.3**	**92.2**	**86.7**	**75.1**	85.7	**89.1**
System (*w/o pre-identified predicates*)	**End-to-end**	**CA**$_{id}$	**CZ**$_{id}$	**CZ**$_{ood}$	**DE**$_{id}$	**DE**$_{ood}$	**ES**$_{id}$	**ZH**$_{id}$
(Syntax-agnostic)								
Li et al. [19]†	n	83.5	89.4	89.2	60.1	43.2	83.0	81.5
This work†	y	**84.6**	**92.4**	**91.3**	**62.0**	**46.1**	**84.5**	**82.6**
Li et al. [19]† + $\text{BERT}_{\text{fine-tuned}}$	n	**85.8**	90.9	90.8	67.2	41.5	**85.8**	85.7
This work† + BERT	y	85.2	**92.2**	**91.3**	**68.5**	**48.8**	85.0	**86.6**

contextualized word embeddings has no effect without given predicates and is harmful with pre-identified predicates, probably meaning that a task-specific fine-tuning would be helpful in that case.

Finally, it is worth mentioning that, to the best of our knowledge, our proposal is the first end-to-end system that provides scores for full multilingual SRL (without given predicates) on CoNLL-2009 datasets, since [19] (the only graph-based model included in that setting) adopts a pipeline strategy. Moreover, our model is the best-performing approach among truly syntax-agnostic SRL systems, which do not exploit PoS tag embeddings. Lastly, we do not fine-tune hyperparameters for individual languages, suggesting that the presented approach is robust and can be directly applied to other languages.

4.4. Ablation study

The previous section has already shown as BERT-based embeddings substantially boost our model accuracy; however, we do not know the performance impact of, for instance, beam-search decoding or high-order features provided by the last assigned predicate. Thus, we conduct an ablation study of our neural architecture in order to better understand the contribution of each component in the final accuracy. In particular, we successively remove from the full model: the beam-search decoding, co-parent features (*i.e.*, state configuration representations s_t are generated without the addition of h_j to h_i), lemma embeddings and character embeddings. In Table 6, we can observe that the removal of every component leads to an overall performance degradation; however, character embedding ablation has the largest impact on the performance of our model, resulting in significant drops in F_1 (−1.68), F_1^{pred} (−1.73) and F_1^{arg} scores (−1.68). Lemma embeddings also play an important role in our neural architecture, notably increasing our model performance. In addition, we can also see as the lack of co-parent features improves the accuracy on predicate identification and disambiguation (F_1^{pred}), but penalizes the performance on argument identification and labeling (F_1^{arg}). This means that co-parent features are especially beneficial for argument processing subtasks, which are more complex and have a larger impact on the overall F_1 score (since, except in Czech, there are significantly more arguments than predicates in CoNLL-2009 corpora). Finally, the beam-search decoding with beam size 5 has a minor impact in comparison to the addition of lemma and character embeddings. We think that a further beam-size exploration might probably increase its contribution to the final accuracy.

4.5. Time complexity

The full time complexity of best-performing graph-based models [19,24] is $O(n^3)$ due to the leverage of higher-order information. We will prove that our approach is more efficient, being $O(n^2)$ its overall expected worst-case running time for the range of data tested in our experiments.

Being n the sentence length, a general directed graph can have at most $\Theta(n^2)$ edges, requiring our transition system $O(n^2)$ actions to build it in the worst case (*i.e.*, n Shift transitions for processing all words and n Arc actions per word for assigning all its heads). Nevertheless, predicate–argument graphs from CoNLL-2009 datasets can be produced with $O(n)$ transitions. In order to prove that, we need to determine the complexity of the proposed transition system in practice. This can be done by examining, for each sentence, how the predicted transition sequence length varies as a function of sentence length [64]. Fig. 3 graphically shows the relation between the number of predicted transitions and the number of words for every sentence from CoNLL-2009 development splits. We can clearly observe a linear relationship across all languages, which means that the number of Arc transitions required per word is notably low and behaves like a constant in the represented linear function. This behavior is supported by the fact that, due to the significant amount of unattached words, there are substantially less

Table 6
Overall precision, recall and F_1 scores, as well as specific F_1 scores measured only on predicate identification+disambiguation (F_1^{pred}) and argument identification+labeling (F_1^{arg}) subtasks on the CoNLL-2009 English development split under the *w/o pre-identified predicates* setup.

System	P	R	F_1	F_1^{pred}	F_1^{arg}
Full model	85.36	85.89	85.63	90.34	83.46
- Beam search	84.69	86.22	$85.44^{(-0.19)}$	$90.20^{(-0.14)}$	$83.26^{(-0.20)}$
- Co-parent features	84.47	86.13	$85.29^{(-0.15)}$	$90.45^{(+0.25)}$	$82.91^{(-0.35)}$
- Lemma embeddings	82.65	86.23	$84.40^{(-0.89)}$	$89.77^{(-0.68)}$	$81.96^{(-0.95)}$
- Character embeddings	80.94	84.58	$82.72^{(-1.68)}$	$88.04^{(-1.73)}$	$80.28^{(-1.68)}$

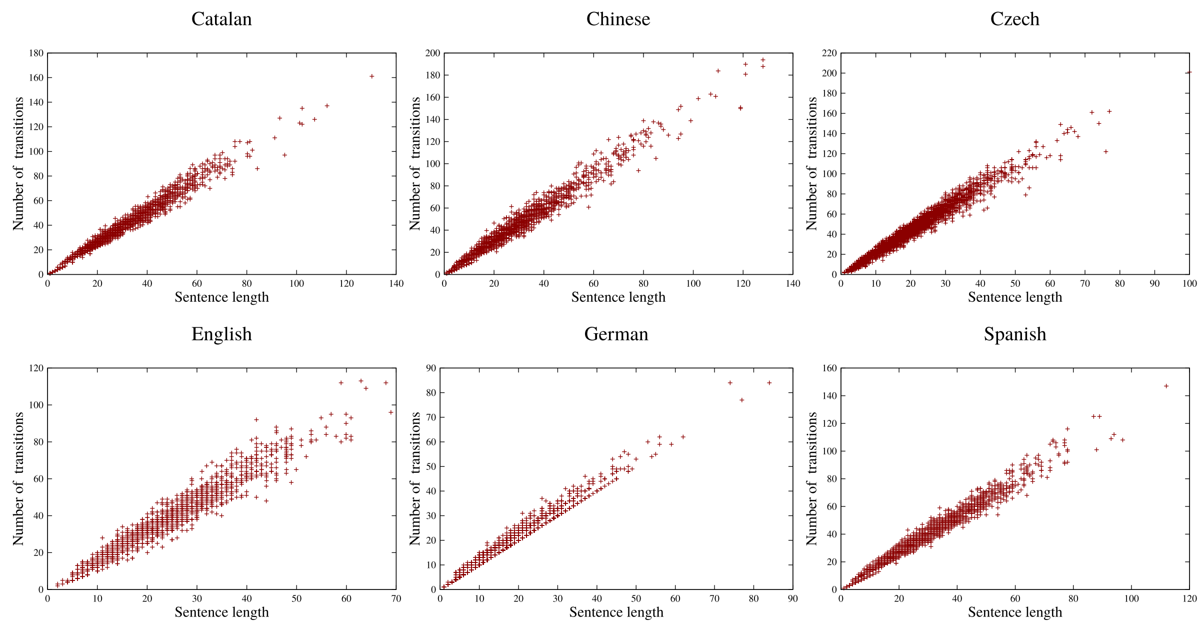

Fig. 3. Number of transitions predicted by our model relative to the sentence length for CoNLL-2009 development sets.

predicate–argument edges than words in graphs from CoNLL-2009 data, being the average ratio of edges per word in a sentence less than 1 in practically all training sets: 0.32 in Catalan, 0.53 in Chinese, 0.08 in German, 0.56 in English and 0.34 in Spanish. The exception is observed in graphs from the Czech dataset, where we have more than one edge per word on average (1.15). From this information, we can state that every sentence from CoNLL-2009 corpora (except Czech) can be processed with $2n$ transitions at most (*i.e.*, n Shift actions plus n Arc transitions); and we will require $3n$ transitions in the worst case for generating graphs from Czech (*i.e.*, n Shift transitions plus $2n$ Arc actions). In both cases, the resulting number of transitions is linear.

Finally, the time complexity of our approach not only comes from the transition system: for predicting each transition, the attention vector α_t must be computed over the whole input sentence in $O(n)$ time. Consequently, the overall time complexity of the proposed SRL system on CoNLL-2009 corpora is $O(n^2)$.

5. Conclusions

In this article, we propose the first syntax-agnostic transition-based approach for full end-to-end SRL. This exclusively relies on raw text as input, neither requiring syntactic trees nor resorting to external models to accomplish any of the SRL subtasks. In addition, we prove that our technique is more efficient in practice ($O(n^2)$) than best-performing graph-based models ($O(n^3)$). Thanks to these advantageous features, it can be easily applied in real-world applications and low-resource languages, where, for instance, syntactic information is scarce.

We not only extensively evaluate our model on in-domain and out-of-domain CoNLL-2009 corpora under the full SRL setting, but additionally adapt our approach to handle gold predicate information in order to perform a fair comparison against the vast majority of previous methods, which do not address predicate identification. While our single-model proposal obtains competitive accuracies on the CoNLL-2009 English data, it excels in the remaining five languages, achieving a strong performance across in-domain and out-of-domain test sets as well as high-resource and low-resource languages.

Lastly, although our transition-based SRL system is robust and accurate, it is outperformed by graph-based models that either fine-tune deep contextualized word embeddings or use additional syntactic information. Therefore, while we think that leveraging syntactic trees makes SRL systems less cost-effective and more dependent on high-resource languages, our model can benefit from syntax to further improve its performance. And, additionally, we could also perform a task-specific fine-tuning of BERT-based embeddings to obtain substantial accuracy gains in low-resource languages.

CRediT authorship contribution statement

Daniel Fernández-González: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Source code available at https://github.com/danifg/SRLPointer.

Acknowledgments

We acknowledge the European Research Council (ERC), which has funded this research under the European Union's Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150), ERDF/MICINN-AEI (SCANNER-UDC, PID2020-113230RB-C21), Xunta de Galicia, Spain (ED431C 2020/11), and Centro de Investigación de Galicia "CITIC", funded by Xunta de Galicia, Spain and the European Union (ERDF - Galicia 2014–2020 Program), by grant ED431G 2019/01.

References

[1] C. Shi, S. Liu, S. Ren, S. Feng, M. Li, M. Zhou, X. Sun, H. Wang, Knowledge-based semantic embedding for machine translation, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 2245–2254, http://dx.doi.org/10.18653/v1/P16-1212, URL: https://aclanthology.org/P16-1212.

[2] R. Wang, H. Zhao, S. Ploux, B.-L. Lu, M. Utiyama, A bilingual graph-based semantic model for statistical machine translation, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI '16, AAAI Press, 2016, pp. 2950–2956.

[3] D. Marcheggiani, J. Bastings, I. Titov, Exploiting semantics in neural machine translation with graph convolutional networks, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 486–492, http://dx.doi.org/10.18653/v1/N18-2078, URL: https://aclanthology.org/N18-2078.

[4] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, Textual inference and meaning representation in human robot interaction, in: Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora, Trento, Italy, 2013, pp. 65–69, URL: https://aclanthology.org/W13-3820.

[5] W.-t. Yih, M. Richardson, C. Meek, M.-W. Chang, J. Suh, The value of semantic parse labeling for knowledge base question answering, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 201–206, http://dx.doi.org/10.18653/v1/P16-2033, URL: https://aclanthology.org/P16-2033.

[6] C. Zheng, P. Kordjamshidi, SRLGRN: Semantic role labeling graph reasoning network, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 8881–8891, http://dx.doi.org/10.18653/v1/2020.emnlp-main.714, URL: https://aclanthology.org/2020.emnlp-main.714.

[7] K. Xu, H. Tan, L. Song, H. Wu, H. Zhang, L. Song, D. Yu, Semantic role labeling guided multi-turn dialogue rewriter, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 6632–6639, http://dx.doi.org/10.18653/v1/2020.emnlp-main.537, URL: https://aclanthology.org/2020.emnlp-main.537.

[8] Z. Zhang, Y. Wu, Z. Li, H. Zhao, Explicit contextual semantics for text comprehension, in: Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation, PACLIC 33, 2019.

[9] L. He, K. Lee, O. Levy, L. Zettlemoyer, Jointly predicting predicates and arguments in neural semantic role labeling, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 364–369, http://dx.doi.org/10.18653/v1/P18-2058, URL: https://aclanthology.org/P18-2058.

[10] E. Strubell, P. Verga, D. Andor, D. Weiss, A. McCallum, Linguistically-informed self-attention for semantic role labeling, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 5027–5038, http://dx.doi.org/10.18653/v1/D18-1548, URL: https://aclanthology.org/D18-1548.

[11] Y. Zhang, Q. Xia, S. Zhou, Y. Jiang, G. Fu, M. Zhang, Semantic role labeling as dependency parsing: Exploring latent tree structures inside arguments, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 4212–4227, URL: https://aclanthology.org/2022.coling-1.370.

[12] M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, J. Nivre, The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies, in: CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning, Coling 2008 Organizing Committee, Manchester, England, 2008, pp. 159–177, URL: https://aclanthology.org/W08-2121.

[13] J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M.A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, Y. Zhang, The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 1–18, URL: https://aclanthology.org/W09-1201.

[14] M. Roth, M. Lapata, Neural semantic role labeling with dependency path embeddings, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1192–1202, http://dx.doi.org/10.18653/v1/P16-1113, URL: https://aclanthology.org/P16-1113.

[15] D. Marcheggiani, A. Frolov, I. Titov, A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling, in: Proceedings of the 21st Conference on Computational Natural Language Learning, CoNLL 2017, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 411–420, http://dx.doi.org/10.18653/v1/K17-1041, URL: https://aclanthology.org/K17-1041.

[16] S. He, Z. Li, H. Zhao, H. Bai, Syntax for semantic role labeling, to be, or not to be, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2061–2071, http://dx.doi.org/10.18653/v1/P18-1192, URL: https://aclanthology.org/P18-1192.

[17] R. Cai, M. Lapata, Syntax-aware semantic role labeling without parsing, Trans. Assoc. Comput. Linguist. 7 (2019) 343–356, http://dx.doi.org/10.1162/tacl_a_00272, URL: https://aclanthology.org/Q19-1022.

[18] S. He, Z. Li, H. Zhao, Syntax-aware multilingual semantic role labeling, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5350–5359, http://dx.doi.org/10.18653/v1/D19-1538, URL: https://aclanthology.org/D19-1538.

[19] Z. Li, H. Zhao, R. Wang, K. Parnow, High-order semantic role labeling, in: Findings of the Association for Computational Linguistics, EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 1134–1151, http://dx.doi.org/10.18653/v1/2020.findings-emnlp.102, URL: https://aclanthology.org/2020.findings-emnlp.102.

[20] S. Conia, R. Navigli, Bridging the gap in multilingual semantic role labeling: A language-agnostic approach, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 1396–1410, http://dx.doi.org/10.18653/v1/2020.coling-main.120, URL: https://aclanthology.org/2020.coling-main.120.

[21] J. Kasai, D. Friedman, R. Frank, D. Radev, O. Rambow, Syntax-aware neural semantic role labeling with supertags, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 701–709, http://dx.doi.org/10.18653/v1/N19-1075, URL: https://aclanthology.org/N19-1075.

[22] Z. Li, H. Zhao, S. He, J. Cai, Syntax role for neural semantic role labeling, Comput. Linguist. 47 (3) (2021) 529–574, http://dx.doi.org/10.1162/coli_a_00408, URL: https://aclanthology.org/2021.cl-3.17.

[23] J. Cai, S. He, Z. Li, H. Zhao, A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2753–2765, URL: https://aclanthology.org/C18-1233.

[24] S. Zhou, Q. Xia, Z. Li, Y. Zhang, Y. Hong, M. Zhang, Fast and accurate end-to-end span-based semantic role labeling as word-based graph parsing, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 4160–4171, URL: https://aclanthology.org/2022.coling-1.365.

[25] X. Ma, Z. Hu, J. Liu, N. Peng, G. Neubig, E. Hovy, Stack-pointer networks for dependency parsing, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1403–1414, http://dx.doi.org/10.18653/v1/P18-1130, URL: https://aclanthology.org/P18-1130.

[26] D. Fernández-González, C. Gómez-Rodríguez, Left-to-right dependency parsing with pointer networks, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short

Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 710–716, http://dx.doi.org/10.18653/v1/N19-1076, URL: https://aclanthology.org/N19-1076.

[27] D. Fernández-González, C. Gómez-Rodríguez, Transition-based semantic dependency parsing with pointer networks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7035–7046, http://dx.doi.org/10.18653/v1/2020.acl-main.629, URL: https://aclanthology.org/2020.acl-main.629.

[28] D. Fernández-González, C. Gómez-Rodríguez, Multitask pointer network for multi-representational parsing, Knowl.-Based Syst. 236 (2022) 107760, http://dx.doi.org/10.1016/j.knosys.2021.107760, URL: https://www.sciencedirect.com/science/article/pii/S0950705121009849.

[29] J.D. Choi, M. Palmer, Transition-based semantic role labeling using predicate argument clustering, in: Proceedings of the ACL 2011 Workshop on Relational Models of Semantics, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 37–45, URL: https://aclanthology.org/W11-0906.

[30] H. Fei, M. Zhang, B. Li, D. Ji, End-to-end semantic role labeling with neural transition-based model, in: Proceedings of the AAAI, 2021, pp. 12803–12811.

[31] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, in: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 28, Curran Associates, Inc., 2015, pp. 2692–2700, URL: http://papers.nips.cc/paper/5866-pointer-networks.pdf.

[32] H. Zhao, W. Chen, J. Kazama, K. Uchimoto, K. Torisawa, Multilingual dependency learning: Exploiting rich features for tagging syntactic and semantic dependencies, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 61–66, URL: https://aclanthology.org/W09-1209.

[33] A. Björkelund, L. Hafdell, P. Nugues, Multilingual semantic role labeling, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 43–48, URL: https://aclanthology.org/W09-1206.

[34] T. Dozat, C.D. Manning, Deep Biaffine attention for neural dependency parsing, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017, URL: https://openreview.net/forum?id=Hk95PK9le.

[35] Z. Li, S. He, H. Zhao, Y. Zhang, Z. Zhang, X. Zhou, X. Zhou, Dependency or span, end-to-end uniform semantic role labeling, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19, AAAI Press, 2019, http://dx.doi.org/10.1609/aaai.v33i01.33016730.

[36] X. Chen, C. Lyu, I. Titov, Capturing argument interaction in semantic role labeling with capsule networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5415–5425, http://dx.doi.org/10.18653/v1/D19-1544, URL: https://aclanthology.org/D19-1544.

[37] C. Lyu, S.B. Cohen, I. Titov, Semantic role labeling with iterative structure refinement, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1071–1082, http://dx.doi.org/10.18653/v1/D19-1099, URL: https://aclanthology.org/D19-1099.

[38] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 1556–1566, http://dx.doi.org/10.3115/v1/P15-1150, URL: https://aclanthology.org/P15-1150.

[39] X. Wang, J. Huang, K. Tu, Second-order semantic dependency parsing with end-to-end neural networks, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4609–4618, http://dx.doi.org/10.18653/v1/P19-1454, URL: https://aclanthology.org/P19-1454.

[40] H. He, J. Choi, Establishing strong baselines for the new decade: Sequence tagging, syntactic and semantic parsing with BERT, in: The Thirty-Third International Flairs Conference, 2020.

[41] S. Oepen, M. Kuhlmann, Y. Miyao, D. Zeman, D. Flickinger, J. Hajič, A. Ivanova, Y. Zhang, SemEval 2014 task 8: Broad-coverage semantic dependency parsing, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEVal 2014, Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 63–72, http://dx.doi.org/10.3115/v1/S14-2008, URL: https://aclanthology.org/S14-2008.

[42] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1064–1074, http://dx.doi.org/10.18653/v1/P16-1101, URL: https://aclanthology.org/P16-1101.

[43] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423, URL: https://aclanthology.org/N19-1423.

[44] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[45] M.P. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated Corpus of English: The Penn treebank, Comput. Linguist. 19 (1993) 313–330.

[46] W. Francis, H. Kucera, Frequency Analysis of English Usage: Lexicon and Usage, Houghton Mifflin, 1982, URL: https://books.google.es/books?id=LrbPAQAACAAJ.

[47] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162, URL: https://aclanthology.org/D14-1162.

[48] W. Ling, C. Dyer, A.W. Black, I. Trancoso, Two/too simple adaptations of Word2Vec for syntax problems, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 1299–1304, http://dx.doi.org/10.3115/v1/N15-1142, URL: https://www.aclweb.org/anthology/N15-1142.

[49] B. Chan, S. Schweter, T. Möller, German's next language model, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 6788–6796, http://dx.doi.org/10.18653/v1/2020.coling-main.598, URL: https://aclanthology.org/2020.coling-main.598.

[50] M. Arkhipov, M. Trofimova, Y. Kuratov, A. Sorokin, Tuning multilingual transformers for language-specific named entity recognition, in: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, Association for Computational Linguistics, Florence, Italy, 2019, pp. 89–93, http://dx.doi.org/10.18653/v1/W19-3712, URL: https://aclanthology.org/W19-3712.

[51] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained BERT model and evaluation data, in: PML4DC At ICLR 2020, 2020.

[52] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, URL: http://arxiv.org/abs/1412.6980, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[53] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML '13, JMLR.org, 2013, pp. III–1310–III–1318.

[54] Y. Gal, Z. Ghahramani, A theoretically grounded application of dropout in recurrent neural networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS '16, Curran Associates Inc., Red Hook, NY, USA, 2016, pp. 1027–1035.

[55] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237, http://dx.doi.org/10.18653/v1/N18-1202, URL: https://aclanthology.org/N18-1202.

[56] J. Zhou, Z. Li, H. Zhao, Parsing all: Syntax and semantics, dependencies and spans, in: Findings of the Association for Computational Linguistics, EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 4438–4449, http://dx.doi.org/10.18653/v1/2020.findings-emnlp.398, URL: https://aclanthology.org/2020.findings-emnlp.398.

[57] K. Munir, H. Zhao, Z. Li, Adaptive convolution for semantic role labeling, IEEE/ACM Trans. Audio Speech Lang. Proc. 29 (2021) 782–791, http://dx.doi.org/10.1109/TASLP.2020.3048665.

[58] T. Lei, Y. Zhang, L. Màrquez, A. Moschitti, R. Barzilay, High-order low-rank tensors for semantic role labeling, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 1150–1160, http://dx.doi.org/10.3115/v1/N15-1121, URL: https://aclanthology.org/N15-1121.

[59] N. FitzGerald, O. Täckström, K. Ganchev, D. Das, Semantic role labeling with neural network factors, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 960–970, http://dx.doi.org/10.18653/v1/D15-1112, URL: https://aclanthology.org/D15-1112.

[60] D. Marcheggiani, I. Titov, Encoding sentences with graph convolutional networks for semantic role labeling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1506–1515, http://dx.doi.org/10.18653/v1/D17-1159, URL: https://aclanthology.org/D17-1159.

[61] Z. Li, S. He, J. Cai, Z. Zhang, H. Zhao, G. Liu, L. Li, L. Si, A unified syntax-aware framework for semantic role labeling, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2401–2411, http://dx.doi.org/10.18653/v1/D18-1262, URL: https://aclanthology.org/D18-1262.

[62] R. Cai, M. Lapata, Semi-supervised semantic role labeling with cross-view training, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1018–1027, http://dx.doi.org/10.18653/v1/D19-1094, URL: https://aclanthology.org/D19-1094.

[63] P. Mulcaire, S. Swayamdipta, N.A. Smith, Polyglot semantic role labeling, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 667–672, http://dx.doi.org/10.18653/v1/P18-2106, URL: https://aclanthology.org/P18-2106.

[64] S. Kübler, R.T. McDonald, J. Nivre, Dependency Parsing, in: Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2009, http://dx.doi.org/10.2200/S00169ED1V01Y200901HLT002.