Comparing Span Extraction Methods for Semantic Role Labeling

Zhisong Zhang, Emma Strubell, Eduard Hovy
Language Technologies Institute, Carnegie Mellon University
zhisongz@cs.cmu.edu, strubell@cmu.edu, hovy@cmu.edu

Abstract

In this work, we empirically compare span extraction methods for the task of semantic role labeling (SRL). While recent progress incorporating pre-trained contextualized representations into neural encoders has greatly improved SRL F1 performance on popular benchmarks, the potential costs and benefits of structured decoding in these models have become less clear. With extensive experiments on PropBank SRL datasets, we find that more structured decoding methods outperform BIO-tagging when using static (word type) embeddings across all experimental settings. However, when used in conjunction with pre-trained contextualized word representations, the benefits are diminished. We also experiment in cross-genre and cross-lingual settings and find similar trends. We further perform speed comparisons and provide analysis on the accuracy-efficiency trade-offs among different decoding methods.

1 Introduction

Semantic role labeling (SRL) is a core natural language processing (NLP) task that aims to identify predicate-argument structures in text (Gildea and Jurafsky, 2002; Palmer et al., 2010). Following the neural encoder-decoder paradigm, we can view an SRL model as combining an encoder, which builds hidden representations for the input words, with a decoder, which extracts the argument spans based on the encoded representations. While recent SRL models achieve high performance on popular benchmarks (Zhou and Xu, 2015; He et al., 2017; Tan et al., 2018; Strubell et al., 2018; Shi and Lin, 2019), most improvements come from better neural encoders, such as the Transformer (Vaswani et al., 2017) and pre-trained contextualized word representations, such as BERT (Devlin et al., 2019). However, influence on end-task performance due to the choice of decoder has become less clear.

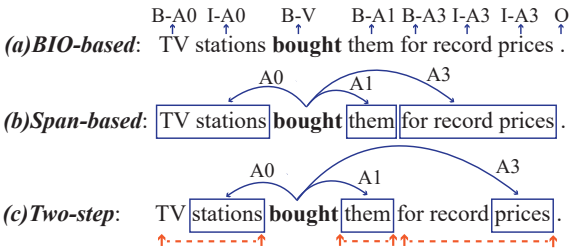

Figure 1: Illustration of decoding methods explored in this work. For the predicate "**bought**", we identify argument spans by: (a) BIO-based sequence labeling; (b) direct span-based extraction; (c) two-step approach: first identifying head words, then expanding to full spans by deciding left and right boundaries.

In this work, we perform an empirical investigation of different decoding methods for span extraction, as illustrated in Figure 1. The most common strategy casts the task as a sequence labeling problem using the BIO-tagging scheme (Zhou and Xu, 2015; He et al., 2017; Tan et al., 2018; Strubell et al., 2018; Shi and Lin, 2019). While this approach is simple, it does not directly model the arguments at the span level. Alternatively, the span-based method directly builds representations for all possible[1] spans and selects among them (He et al., 2018a; Ouchi et al., 2018). Though this approach is straightforward for explicitly modeling span-level information, composing a representation for every span can lead to higher computational cost. Inspired by dependency-based SRL (Surdeanu et al., 2008; Hajič et al., 2009), a third option first identifies a head word then decides the span boundaries. This two-step strategy has been explored in previous work on information extraction (Peng et al., 2015; Lin et al., 2019; Zhang et al., 2020), and we apply it here to SRL. Compared with the sequential BIO-tagger, the latter two approaches more directly model the argument span structures; we thus refer

[1] Up to a fixed length, decided as a hyperparameter.

Proceedings of the 5th Workshop on Structured Prediction for NLP, pages 67–77
August 1–6, 2021.

to them as more *structured* decoders.

We perform careful comparisons of these decoding methods upon the same encoding backbone, based on a deep Transformer encoder. We first experiment in the standard fully-supervised settings on English PropBank datasets (CoNLL-2005 and CoNLL-2012). The results show that more structured decoders, especially the two-step approach with syntactic guidance, consistently perform better than BIO-tagging when using static word embeddings. However, if including strong contextualized BERT embeddings, the benefits of more structured decoding are diminished and the simplest BIO-tagging method performs well across different experimental settings. Error analysis shows that contextualized embeddings help in deciding span boundaries. Furthermore, we explore cross-genre and cross-lingual settings on the CoNLL-2012 datasets, and find similar trends. Finally, we perform speed comparisons and analyze the accuracy-efficiency trade-offs among different decoding methods.

2 Model

For a given predicate,[2] SRL aims to extract all argument spans and assign them role labels. To model this task, we follow the neural encoder-decoder paradigm: the encoder produces hidden representations for the input words, upon which the decoder decides the structured outputs. All our models adopt the same encoding architecture: a deep Transformer encoder (Vaswani et al., 2017), which has been shown effective for SRL (Tan et al., 2018; Strubell et al., 2018). For a given input sequence of words $\{w_1, \ldots, w_n\}$, we obtain their contextualized representations $\{h_1, \ldots, h_n\}$ from the encoder. Upon these, we stack different decoders to extract the argument spans corresponding to different extraction strategies, which will be described in the following.

2.1 BIO-based

Since argument spans do not overlap in the datasets we explore, the BIO-tagging scheme (Ramshaw and Marcus, 1999) can be utilized to extract them, casting SRL as a sequence labeling problem.

For each word, we feed its representation h to a multi-layer perceptron (MLP) based scorer, which assigns the scores of the BIO tags. Assuming that we have k possible argument roles in the output space, each of them will have its "B-" and "I-" tags. Together with the "O" (NIL) tag, the tagging space has a dimension of $2k + 1$.

Furthermore, we consider the option of adopting a standard linear-chain conditional random field (CRF; Lafferty et al., 2001) to model pairwise tagging transitions. If adopting the CRF (BIO w/ CRF), we train the model with sequence-level negative log likelihood and use the Viterbi algorithm for inference. If not using the CRF (BIO w/o CRF), we simply use tag-level cross entropy as the learning objective and perform argmax greedy decoding at inference time, following Tan et al. (2018).

2.2 Span-based

In the span-based method, we build neural representations for all candidate spans and directly select and assign role labels (or NIL). Following He et al. (2018a), for a span a, we compose its representation from start and end points, soft head-word vectors and span width features by concatenation:

$$\mathbf{g}(a) = [h_{start(a)}, h_{end(a)}, \text{soft}(a), \text{width}(a)]$$

Here, $\text{soft}(a)$ denotes a soft-head representation obtained from an attention mechanism:

$$\text{soft}(a) = \sum_{start(a) \le i \le end(a)} \text{att}(i, a) h_i$$

$$\text{att}(i, a) = \frac{w_{att}^T h_i}{\sum_{start(a) \le i' \le end(a)} w_{att}^T h_{i'}}$$

and $\text{width}(a)$ denotes a width embedding corresponding to the span size (width).

All valid candidate spans are first assigned an unlabeled score, using an MLP scorer. This unary score is then used as the criterion for beam pruning to reduce the computational costs of full labeling. Since each predicate will not have too many arguments (most have less than 5), we adopt a fixed beam size of 10. We also limit the maximum width of candidate spans to 30, which covers around 99% of the cases. Surviving candidates are further assigned label scores with another MLP scorer, with which we decide output arguments.

2.3 Two-step

In this approach, we decompose the problem into two steps: head-selection and boundary-decision. In the first step, each individual word is directly scored for argument labels (or NIL). We again adopt an MLP classifier to obtain the probability

[2] In this work, we focus on argument extraction and assume given predicates.

that a word can be the head of an argument with label r (r can be NIL). The non-NIL labeled words are selected as the head words of the arguments. Since the annotations usually do not contain head words for the argument spans, we further consider two strategies to provide supervision for training:

HeadSyntax A straightforward method is to adopt guidance from syntax. Following dependency-style SRL (Surdeanu et al., 2008; Hajič et al., 2009), we use syntactic dependency parse trees and select the highest word (the one that is closest to the root) in the span as the head. In training, we only assign the argument role to the syntactic head word, and all other words in the span get a label of NIL.

HeadAuto In this strategy, all words in an argument span can be considered as the potential head word. We adopt the *bag loss* from Lin et al. (2019) to train the model to automatically identify head words. Specifically, for a word w_i inside an argument span a which has the role r, the loss is computed as:

$$\begin{aligned}\text{Loss}(w_i) &= \delta_i \cdot [-\log p(r|h_i)] \\ &\quad + (1-\delta_i) \cdot [-\log p(\text{NIL}|h_i)] \\ \delta_i &= \frac{p(r|h_i)}{\max_{start(a) \le j \le end(a)} p(r|h_j)}\end{aligned}$$

Here, words that are more indicative for the argument will be assigned higher probabilities to the argument role. This will give them larger loss weights (δ) and thus further encourage them to be the heads. In this way, the head words are decided automatically by the model.

In the second step, we determine span boundaries for these head words. Here we adopt the span selection method from extractive question answering (Wang and Jiang, 2016; Devlin et al., 2019) using two classifiers to decide the start and end words ($[s, e]$) of a span:

$$\begin{aligned}p(s,e) &= p_{start}(s) \cdot p_{end}(e) \\ p_{start}(s) &= \frac{\exp \text{score}_{start}(h'_s)}{\sum_i \exp \text{score}_{start}(h'_i)} \\ p_{end}(e) &= \frac{\exp \text{score}_{end}(h'_e)}{\sum_i \exp \text{score}_{end}(h'_i)}\end{aligned}$$

Here, we first add indicator embeddings to the head word's encoder representations to mark its positions, and then stack one self-attention layer to obtain head-word-aware representations for the input sequence: $\{h'_1, \cdots, h'_n\}$. We further introduce two linear scorers to assign the start and end scores for each word, which are further normalized across the input sequence. For training, the objective is minimizing the sum of negative log-likelihoods of picking the correct start and end positions. When decoding, we select the maximum scoring span whose boundaries s and e satisfy $s \le e$.

We observe that at inference time, sometimes different head words may expand to overlapping spans, which do not appear in the datasets we explore. To deal with this, we adopt a greedy post-processing procedure to remove overlapping argument spans: iterating through all argument spans ranked by model score and only keeping the ones that do not overlap with previous surviving ones.

3 Experiments

3.1 Settings

Data The models are evaluated on standard PropBank datasets from the CoNLL-2005 shared task (Carreras and Màrquez, 2005) and the CoNLL-2012 subset of OntoNotes 5.0 (Pradhan et al., 2013). Table 1 lists the relevant statistics. For CoNLL-2005, we follow the splits from the CoNLL-2005 shared task.[3] For the English part of CoNLL-2012, we adopt the data from Pradhan et al. (2013)[4] but follow the splits of the CoNLL-2012 shared task.[5] For the Chinese part of CoNLL-2012, we directly utilize those provided by the CoNLL-2012 shared task. For evaluation, we adopt the standard evaluation script of `srl-eval.pl`.[6] For the "HeadSyntax" method that requires dependency trees, we convert the original constituencies to Universal Dependencies (Nivre et al., 2020) using Stanford CoreNLP (Manning et al., 2014) version 4.1.0. Notice that we only need syntactic information to be provided during training, since the model predicts head words itself at test time.

Input Features and Encoder For fair comparison, we adopt the same input features, deep Transformer-based encoders and training schemes across all experiments. We consider two types of word features: static word embeddings and

[3] https://www.cs.upc.edu/~srlconll/

[4] https://cemantix.org/data/ontonotes.html

[5] https://conll.cemantix.org/2012/

[6] https://www.cs.upc.edu/~srlconll/soft.html

	CoNLL 2005				CoNLL 2012 (English)			CoNLL 2012 (Chinese)		
	Train	Dev	Test	Brown	Train	Dev	Test	Train	Dev	Test
Sent.	39.8k	1.3k	2.4k	0.4k	75.2k	9.6k	9.5k	36.5k	6.1k	4.5k
Pred.	90.8k	3.2k	5.3k	0.8k	188.9k	23.9k	24.5k	117.1k	16.6k	15.0k
Arg.	333.7k	11.7k	19.6k	3.0k	622.5k	78.1k	80.2k	365.3k	51.0k	46.7k

Table 1: Statistics of the datasets: Number of sentences (Sent.), predicates (Pred.) and arguments (Arg.).

pre-trained contextualized embeddings[7] from $\text{BERT}_{\text{base}}$. In the English experiments, we adopt fastText[8] embeddings (Mikolov et al., 2018) and frozen features from `bert-base-cased`. In the cross-lingual experiments, we only utilize multi-lingual BERT features from `bert-base-multilingual-cased`. Before feeding the word-level features to the encoder, we concatenate them and apply a linear layer to project them to the encoding dimension. We further add indicator embeddings to let the model be aware of the positions of the predicates. For both cases of static embedding and BERT features, we adopt a 10-layer Transformer module as the encoder. The head number, model dimension and feed-forward dimension are set to 8, 512 and 1024, respectively. In addition, we adopt relative positional encodings for the Transformer (Shaw et al., 2018) since we found slightly better performance in preliminary experiments.

Training We use the Adam optimizer (Kingma and Ba, 2014) for training. The learning rate is linearly increased towards 2e-4 within the first 8k steps as warm up. After this, we decay the learning rate by 0.75 each time the performance on the development set does not increase for 10 checkpoints. We train the model for a maximum of 150k steps and do validation every 1k steps to select the best model. One model contains around 40M parameters (excluding BERT). For each update, the batch size is around 4096 tokens. We apply dropout rates of 0.2 to the hidden layers. For models using static embeddings, we further replace input words by a special UNK token with a probability of 0.5 if it appears less than 3 times in the training set. At test time, a word is represented by UNK if it is not found in the collection of static word embeddings. All the experiments are run with our own implementation[9]. All the models are trained and evaluated on one TITAN-RTX GPU, and training one model takes around 1 day in our environment.

[7]We concatenate layer 7, 8 and 9 of BERT hidden representations. For words that are split into sub-tokens, we utilize the representations of the first sub-token.

[8]`https://fasttext.cc/docs/en/english-vectors.html`

Model	WSJ	Brown	OntoNotes
He et al. (2018a)	87.4	80.4	85.5
Ouchi et al. (2018)	87.6	78.7	86.2
Shi and Lin (2019)	**88.8**	82.0	86.5
Ours (BIO w/ CRF)	87.9	**82.1**	**86.6**

Table 2: Comparisons of F1 scores with previous work in the fully-supervised settings (with single model).

3.2 Fully-supervised Experiments

We first experiment in the fully-supervised settings on English data. Table 2 lists the comparisons of our test results (BIO w/ CRF using BERT features) to previous work. Generally our model can obtain comparable results, which verifies the quality of our implementation.

Tables 3 and 4 list our main comparisons on the development and test sets. The overall trends are very similar. For BIO-tagging, incorporating a structured CRF layer is generally helpful, which can improve the F1 scores by around 0.5 points. When not using BERT features, more structured decoders generally perform better than BIO-tagging. With the head word oracles from the syntax trees, "HeadSyntax" performs the best overall. This agrees with Strubell et al. (2018) and Swayamdipta et al. (2018), showing the helpfulness of syntactic information for SRL. However, when utilizing BERT features, the benefits of more structured decoders are diminished and the simple BIO-tagger robustly performs well. It seems that with a powerful encoder, the choice of the decoder plays a smaller role for final performance.

To further investigate this phenomenon, we perform error analysis on the development outputs of "BIO (w/ CRF)" and "HeadSyntax," which are the two that perform the best overall. We group the errors into four categories: "Boundary" denotes that the predicted head words and role labels match

[9]`https://github.com/zzsfornlp/zmsp/`

	CoNLL2005 In-domain (WSJ)			CoNLL 2012 (OntoNotes)		
	P	R	F1	P	R	F1
Without BERT						
BIO (w/o CRF)	83.11	83.89	83.49$_{\pm 0.20}$	81.43	82.75	82.09$_{\pm 0.22}$
BIO (w/ CRF)	83.66	84.27	83.96$_{\pm 0.26}$	82.41	**83.77**	83.09$_{\pm 0.11}$
Span	84.60	83.57	84.08$_{\pm 0.23}$	82.89	83.04	82.96$_{\pm 0.12}$
HeadSyntax	**84.81**	**84.48**	**84.65**$_{\pm 0.18}$	**83.12**	83.42	**83.27**$_{\pm 0.18}$
HeadAuto	84.52	84.38	84.45$_{\pm 0.22}$	82.50	83.16	82.83$_{\pm 0.15}$
With BERT						
BIO (w/o CRF)	86.47	87.50	86.98$_{\pm 0.12}$	85.22	86.94	86.08$_{\pm 0.15}$
BIO (w/ CRF)	86.78	**87.84**	87.31$_{\pm 0.13}$	85.66	**87.19**	**86.42**$_{\pm 0.12}$
Span	86.94	86.76	86.85$_{\pm 0.16}$	85.83	86.37	86.10$_{\pm 0.11}$
HeadSyntax	**87.35**	87.48	**87.41**$_{\pm 0.14}$	**86.04**	86.79	86.41$_{\pm 0.12}$
HeadAuto	87.10	87.67	87.38$_{\pm 0.22}$	85.80	86.75	86.27$_{\pm 0.15}$

Table 3: Development results for the fully-supervised experiments. All the numbers are averaged over 5 runs with different random seeds, standard deviations of F1 scores are also reported.

	CoNLL 2005 In-domain (WSJ)			Out-of-domain (Brown)			CoNLL 2012 (OntoNotes)		
	P	R	F1	P	R	F1	P	R	F1
Without BERT									
BIO (w/o CRF)	84.42	84.94	84.68$_{\pm 0.25}$	73.56	73.03	73.29$_{\pm 0.43}$	81.74	82.98	82.35$_{\pm 0.24}$
BIO (w/ CRF)	85.04	85.35	85.20$_{\pm 0.12}$	74.25	73.92	74.08$_{\pm 0.31}$	82.79	**84.11**	83.44$_{\pm 0.21}$
Span	85.68	84.62	85.14$_{\pm 0.32}$	75.88	74.23	75.05$_{\pm 0.42}$	83.42	83.49	83.46$_{\pm 0.15}$
HeadSyntax	**85.84**	**85.38**	**85.61**$_{\pm 0.11}$	**75.92**	**74.74**	**75.33**$_{\pm 0.58}$	**83.55**	83.82	**83.68**$_{\pm 0.11}$
HeadAuto	85.30	85.17	85.23$_{\pm 0.14}$	74.98	73.85	74.41$_{\pm 0.50}$	83.09	83.71	83.40$_{\pm 0.09}$
With BERT									
BIO (w/o CRF)	87.21	87.95	87.58$_{\pm 0.28}$	81.26	81.79	81.52$_{\pm 0.23}$	85.33	86.97	86.14$_{\pm 0.10}$
BIO (w/ CRF)	87.54	**88.32**	**87.93**$_{\pm 0.16}$	81.91	**82.37**	**82.14**$_{\pm 0.20}$	85.93	**87.32**	**86.62**$_{\pm 0.14}$
Span	87.75	87.33	87.54$_{\pm 0.14}$	81.87	81.60	81.73$_{\pm 0.77}$	85.97	86.26	86.12$_{\pm 0.09}$
HeadSyntax	**87.76**	87.96	87.86$_{\pm 0.08}$	**82.10**	81.60	81.85$_{\pm 0.90}$	**86.17**	86.77	86.47$_{\pm 0.10}$
HeadAuto	87.70	88.15	**87.93**$_{\pm 0.12}$	81.52	81.36	81.44$_{\pm 0.37}$	86.00	86.84	86.42$_{\pm 0.09}$

Table 4: Test results of the fully-supervised experiments. All the results are averaged over five runs with different random seeds, standard deviations of the F1 scores are also reported.

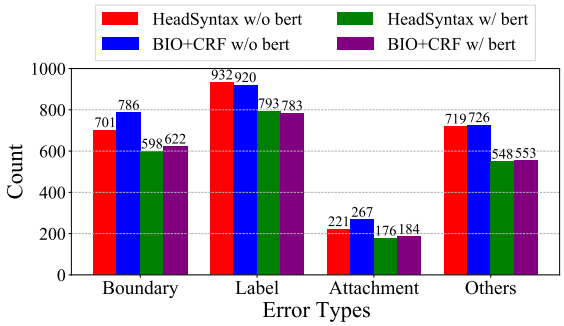

Figure 2: Error breakdown for "BIO" and "HeadSyntax" on the CoNLL-2005 development set.

the gold ones but the span boundaries are incorrect; "Label" denotes that the predicted spans are correct but the role labels are wrong; "Attachment" denotes the errors caused by incorrect phrase attachments, while "Others" denotes the remaining errors, which are other missing and over-predicted arguments. The results are shown in Figure 2. When not using BERT features, the main advantages of "HeadSyntax" over "BIO" are on the "Boundary" and "Attachment" errors, where the former makes 11% fewer "Boundary" and 17% fewer "Attachment" errors. Notice that these two types of errors are closely related to syntax, and they are mainly caused by incorrect phrase boundary predictions. In this way, it seems natural that incorporating syntactic information with head words can be helpful in this scenario. Nevertheless, when utilizing BERT features, these advantages are reduced to a negligible level. This indicates that BERT may provide sufficient information overlapping with syntax to help on boundary decisions.

3.3 Cross-genre Experiments

We further explore English cross-genre settings. We utilize English CoNLL-2012 subsets of

	nw*	bc	bn	mz	pt	tc	wb	Avg.
Without BERT								
BIO (w/o CRF)	77.51$_{\pm 0.17}$	59.91$_{\pm 0.31}$	73.28$_{\pm 0.62}$	71.15$_{\pm 0.37}$	81.03$_{\pm 0.31}$	67.90$_{\pm 0.37}$	72.36$_{\pm 0.07}$	71.88
BIO (w/ CRF)	78.42$_{\pm 0.39}$	60.15$_{\pm 0.40}$	73.97$_{\pm 0.15}$	71.37$_{\pm 0.13}$	81.51$_{\pm 0.36}$	68.72$_{\pm 0.34}$	72.54$_{\pm 0.41}$	72.38
Span	79.08$_{\pm 0.16}$	62.74$_{\pm 0.49}$	74.80$_{\pm 0.30}$	72.77$_{\pm 0.36}$	**82.42**$_{\pm 0.41}$	68.93$_{\pm 0.12}$	74.17$_{\pm 0.15}$	73.56
HeadSyntax	**79.54**$_{\pm 0.37}$	**62.81**$_{\pm 0.58}$	**75.06**$_{\pm 0.25}$	**73.17**$_{\pm 0.32}$	82.10$_{\pm 0.30}$	68.74$_{\pm 0.54}$	**74.82**$_{\pm 0.19}$	**73.75**
HeadAuto	79.04$_{\pm 0.22}$	61.97$_{\pm 0.30}$	74.09$_{\pm 0.25}$	72.56$_{\pm 0.40}$	81.80$_{\pm 0.40}$	**69.25**$_{\pm 0.39}$	73.96$_{\pm 0.19}$	73.24
With BERT								
BIO (w/o CRF)	83.55$_{\pm 0.24}$	73.37$_{\pm 0.51}$	80.02$_{\pm 0.19}$	78.45$_{\pm 0.34}$	87.63$_{\pm 0.19}$	74.89$_{\pm 0.41}$	79.49$_{\pm 0.29}$	79.63
BIO (w/ CRF)	83.73$_{\pm 0.28}$	75.24$_{\pm 0.89}$	80.64$_{\pm 0.15}$	78.75$_{\pm 0.56}$	**87.94**$_{\pm 0.42}$	75.38$_{\pm 0.42}$	79.66$_{\pm 0.39}$	80.19
Span	83.41$_{\pm 0.18}$	74.22$_{\pm 0.89}$	80.85$_{\pm 0.29}$	78.69$_{\pm 0.39}$	87.44$_{\pm 0.16}$	75.05$_{\pm 0.36}$	79.44$_{\pm 0.33}$	79.87
HeadSyntax	**83.96**$_{\pm 0.34}$	**75.98**$_{\pm 0.94}$	**80.88**$_{\pm 0.17}$	**79.36**$_{\pm 0.37}$	87.40$_{\pm 0.25}$	75.12$_{\pm 0.41}$	**80.05**$_{\pm 0.20}$	**80.39**
HeadAuto	83.76$_{\pm 0.28}$	74.98$_{\pm 0.77}$	80.69$_{\pm 0.21}$	79.01$_{\pm 0.27}$	87.33$_{\pm 0.36}$	**75.66**$_{\pm 0.54}$	79.98$_{\pm 0.10}$	80.20

Table 5: F1 scores of the (English) cross-genre experiments (averaged over 5 runs with different random seeds). "*" denotes that models are trained on the "nw*" portion. "Avg." denotes macro average over all genres.

	bc	bn	mz
BIO (w/o CRF)	71.19$_{\pm 0.61}$	77.56$_{\pm 0.68}$	76.63$_{\pm 0.51}$
BIO (w/ CRF)	72.11$_{\pm 0.98}$	76.28$_{\pm 0.61}$	75.87$_{\pm 0.69}$
Span	73.30$_{\pm 1.07}$	79.90$_{\pm 0.58}$	77.90$_{\pm 0.59}$
HeadSyntax	**75.23**$_{\pm 1.00}$	**79.95**$_{\pm 0.49}$	**78.69**$_{\pm 0.41}$
HeadAuto	73.60$_{\pm 0.49}$	78.97$_{\pm 0.53}$	77.60$_{\pm 0.41}$

Table 6: F1 scores of the (English) cross-genre experiments (averaged over 5 runs with different random seeds) on specific genres without excluding auxiliary predicates (with BERT).

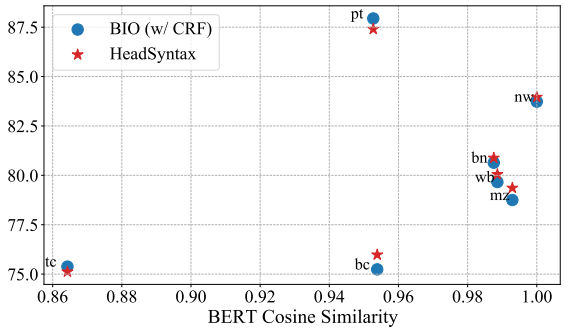

Figure 3: F1 results versus genre similarities according to BERT representations.

OntoNotes and split the corpus according to the genres. There are seven genres, including broadcast conversation (bc), broadcast news (bn), newswire (nw), magazine (mz), pivot (Bible) (pt), telephone conversation (tc) and web (wb) text. The models are trained on the newswire (nw) portion and directly evaluated on portions of all the genres. Table 5 shows the test results. The overall trends are similar to those in the fully-supervised setting. Without BERT, more span-aware structured decoders perform better by more than for 1 point compared to BIO-tagging. After including BERT features, the gaps decrease. Nevertheless, more structured decoders can still perform competitively.

Note that in this setting, we perform evaluations with a correction to an annotation inconsistency that originally favored more structured (direct) decoders. We find that there are inconsistent annotations for the predicates of auxiliary verbs across some genres, we thus exclude them[10] for evaluation. In the genres of "bc", "bn" and "mz", there are many more auxiliary verbs annotated than those in "nw". Interestingly, if not excluding these examples, the more structured decoders perform better than BIO-tagging even with BERT, as shown in Table 6. A possible explanation is that the more structured decoders usually see more negative examples during training and might be more conservative when predicting arguments for these auxiliary verbs, which do not have any arguments. On the contrary, the BIO-tagger tends to over-predict arguments in these cases, leading to worse results. Nevertheless, this phenomenon is only the result of an annotation inconsistency in the dataset and we thus exclude these auxiliary verbs from evaluation in this setting.

We further compare cross-genre results with genre (domain) similarities. Following Aharoni and Goldberg (2020), we obtain similarity scores from target genres to the source genre (nw) by calculating cosine similarity of the centroids of BERT representations. Specifically, we first compute sentence-level representations by average pooling the final hidden vectors with a vanilla BERT, then the genre-level representations are obtained by fur-

[10] We exclude ["be.03", "become.03", "do.01", "have.01"].

	Dev	Test
BIO (w/o CRF)	$56.73_{\pm 0.63}$	$56.18_{\pm 0.61}$
BIO (w/ CRF)	$56.86_{\pm 1.05}$	$56.47_{\pm 0.95}$
Span	$56.61_{\pm 0.51}$	$55.97_{\pm 0.39}$
HeadSyntax	$\mathbf{57.05}_{\pm 0.36}$	$56.48_{\pm 0.34}$
HeadAuto	$\mathbf{57.05}_{\pm 0.59}$	$\mathbf{56.51}_{\pm 0.66}$

Table 7: Unlabeled F1 scores of English→Chinese zero-shot cross-lingual experiments (averaged over five runs with different random seeds).

Gold	[你] [在 纽约 时报 上] 写 了 [一 篇 文章]
Literally	[you] [at New York Times] wrote [an article]
Predicted	[你] [在 纽约 时报 上] 写 [了 一 篇 文章]

Table 8: A typical error of cross-lingual systems. Here, the predicate is the underlined "写"(wrote) and the gold and predicted arguments are presented in [the brackets]. The cross-lingual models wrongly include the extra auxiliary word "了" in the last argument.

ther averaging all sentence-level ones in the corpus. We show the results of "BIO (w/ CRF)" and "HeadSyntax" in Figure 3. Generally, F1 scores on target genres have a weak correlation with genre similarities to the source (Pearson's correlation is 0.45). The outlier "pt" is a special case (biblical text) which mainly contains simple instances.

3.4 Cross-lingual Experiments

We further explore a simple zero-shot cross-lingual setting. We still take the CoNLL-2012 subset of the Ontonotes corpus. The models are trained on the English sets, and then directly applied to the Chinese sets. This time we exclude word embeddings and only use representations from multilingual BERT as the input features, which has shown to be effective for cross-lingual transfer (Wu and Dredze, 2019). Since the Chinese and English PropBanks use different frames, the labeled results might not be directly comparable. We thus perform unlabeled training and evaluate unlabeled argument F1 scores, which reveal how well the models extract argument spans. We simply collapse all the role labels into one special "IsArg" label.

The results are listed in Table 7. The trends are still similar to the previous monolingual experiments with BERT, different decoders obtain similar results, especially considering the deviations of multiple runs. In this setting, the CRF does not help as much as in the case of monolingual experiments. The main reason might be that we are training unlabeled systems, and the main transition that CRF is capturing is "I" after "B", which does not provide too much enhancement. Interestingly, in our preliminary experiments, we also tried labeled training, and found that the CRF is actually harmful, since the distributions of the tag transitions might be different across languages.

We further investigate the systems' outputs and find similar error patterns. Table 8 lists a typical example, where in Chinese the auxiliary word "了" (which denotes perfective aspect[11]) is incorrectly included in the argument. This error is not surprising if considering that in the English training corpus, the predicate verbs usually have directly-following arguments. All extraction methods explored in this work are unlikely to fix such errors without language-specific knowledge.

Decoding	Without BERT	With BERT
BIO (w/o CRF)	$709.8_{\pm 10.6}$	$412.3_{\pm 4.6}$
BIO (w/ CRF)	$497.0_{\pm 4.5}$	$335.1_{\pm 4.3}$
Span	$355.8_{\pm 5.4}$	$261.3_{\pm 3.7}$
HeadSyntax	$561.6_{\pm 5.1}$	$372.8_{\pm 4.5}$
HeadAuto	$454.9_{\pm 7.9}$	$311.0_{\pm 5.8}$

Table 9: Speed comparisons of decoding methods (evaluated by number of sequences per second, averaged over 5 runs, on one TITAN-RTX GPU).

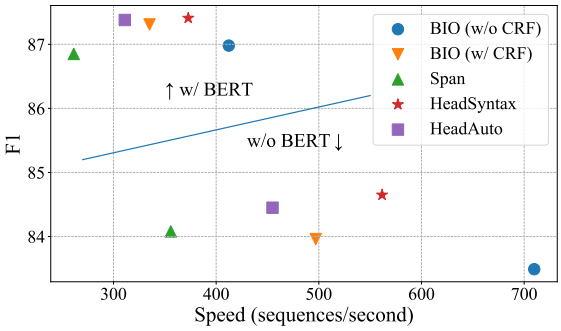

Figure 4: Comparing speed vs. F1 with different decoding methods (on CoNLL05 development set).

3.5 Speed Comparisons

Finally we compare the decoding speed of different extraction methods. Results are shown in Table 9 and we further compare them against F1 scores in Figure 4. Greedy BIO-tagging (w/o CRF) obtains the highest speed. However, this comes with a drop of around 0.5 F1 points without BERT and 0.3 F1 points with BERT. Although the two-step approaches require two decoding steps, they

[11] https://universaldependencies.org/zh/dep/aux_.html

are still efficient thanks to the simplicity of both steps. When trained with syntactic information, this model is the second best in terms of decoding speed. On the other hand, even with beam pruning, the span-based decoder still needs to score a number of span candidates quadratic in the input sequence length, making it less efficient compared to other decoders.

4 Related Work

Argument Extraction Before the incorporation of end-to-end neural models, traditional SRL systems usually depend on input constituency trees to obtain argument candidates (Xue and Palmer, 2004; Màrquez et al., 2008). Although straightforward, this may suffer from error propagation from syntax parsers. Recent neural systems utilize end-to-end models to solve the task. Casting SRL as BIO-based sequence labeling problem is the most common decoding scheme and can obtain impressive results (Zhou and Xu, 2015; He et al., 2017; Tan et al., 2018; Strubell et al., 2018; Shi and Lin, 2019). On the other hand, span-based methods (He et al., 2018a; Ouchi et al., 2018) directly select and label among argument span candidates. This is actually similar to the traditional approaches, though the argument candidates are obtained by the model rather than from input syntax trees. In addition to span-based SRL, the focus of this work, there is another category of dependency-style SRL, which only requires the extraction of head words of argument spans (Surdeanu et al., 2008; Hajič et al., 2009). Inspired by this, for span-based SRL, we can extract argument head words as the first step and then expand to the full spans in a second step. This idea has also been applied in information extraction, such as coreference resolution (Peng et al., 2015), entity detection (Lin et al., 2019) and event argument extraction (Zhang et al., 2020). Another interesting direction is considering the structured constraints of the arguments, including works on integer linear programming (Punyakanok et al., 2004, 2008), dynamic programming (Täckström et al., 2015) and structure-aware tuning (Li et al., 2020).

Syntax and SRL There has been discussion of the relation between syntax and SRL (Gildea and Palmer, 2002; Punyakanok et al., 2008), considering the close connections between these two tasks. Though syntax trees are usually the inputs to traditional SRL systems, some recent works find that syntax-agnostic neural models also work well (Marcheggiani et al., 2017; Cai et al., 2018). Nevertheless, with recent neural models, syntax information has still been found helpful for SRL in various ways, including multi-task learning (Swayamdipta et al., 2018; Strubell et al., 2018), argument pruning (He et al., 2018b), and tree-based modeling (Marcheggiani and Titov, 2017; Li et al., 2018; Marcheggiani and Titov, 2020). In this work, our "HeadSyntax" decoder incorporates syntax in a partial way, utilizing dependency trees to decide the head words in training. This method indeed performs the best overall if only adopting static word embeddings. However, the incorporation of BERT features diminishes the advantages. This indicates that BERT may already cover much of the syntactic (surface) features of the input sentences, as suggested by recent works on BERT interpretation (Goldberg, 2019; Hewitt and Manning, 2019; Tenney et al., 2019; Clark et al., 2019).

Cross-lingual SRL There has also been increasing interest in cross-lingual transfer for SRL, where data transfer and model transfer are the main approaches. Data transfer usually depends on translation and annotation projection to obtain training resources for target languages (Padó and Lapata, 2009; Akbik et al., 2015; Aminian et al., 2019; Fei et al., 2020a; Daza and Frank, 2020). On the other hand, model transfer techniques directly reuse an SRL model trained on source languages to transfer to target languages (Kozhevnikov and Titov, 2013; Fei et al., 2020b), based on common representations. In particular, the recent development of multilingual neural representations, such as multilingual BERT, has been shown to be effective for cross-lingual transfer (Wu and Dredze, 2019; Pires et al., 2019). In this work, we explore a simple zero-shot unlabeled setting for cross-lingual SRL. We leave more explorations on this to future work.

5 Conclusion

In this work, we empirically compare several span extraction methods for SRL. Extensive results show that in fully supervised settings, simple BIO-tagging is a robustly good option when utilizing BERT features. Similar trends are also found in cross-genre and cross-lingual settings. We also analyze the accuracy-efficiency trade-offs for different decoders; although methodologically more complex, two-step approaches are still efficient in decoding. Future work could explore other NLP tasks that require extracting textual spans.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating high quality proposition Banks for multilingual semantic role labeling. 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)*, pages 397–407, Beijing, China. Association for Computational Linguistics.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2019. Cross-lingual transfer of semantic roles: From raw text to semantic roles. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 200–210, Gothenburg, Sweden. Association for Computational Linguistics.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Angel Daza and Anette Frank. 2020. X-SRL: A parallel cross-lingual semantic role labeling dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3904–3914, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Fei, Meishan Zhang, and Donghong Ji. 2020a. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7014–7026, Online. Association for Computational Linguistics.

Hao Fei, Meishan Zhang, Fei Li, and Donghong Ji. 2020b. Cross-lingual semantic role labeling with model transfer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2427–2437.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 239–246, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. 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*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018a. 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)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018b. 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)*, pages 2061–2071, Melbourne, Australia. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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), pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mikhail Kozhevnikov and Ivan Titov. 2013. Cross-lingual transfer of semantic role labeling models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1200, Sofia, Bulgaria. Association for Computational Linguistics.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Tao Li, Parth Anand Jawale, Martha Palmer, and Vivek Srikumar. 2020. Structured tuning for semantic role labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8402–8412, Online. Association for Computational Linguistics.

Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si. 2018. A unified syntax-aware framework for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2411, Brussels, Belgium. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. 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)*, pages 411–420, Vancouver, Canada. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2020. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3915–3928, Online. Association for Computational Linguistics.

Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. 2008. Special issue introduction: Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 12–21, Beijing, China. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1346–1352, Geneva, Switzerland. COLING.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. 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*, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3772–3782, Brussels, Belgium. Association for Computational Linguistics.

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 88–94, Barcelona, Spain. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural 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)*, pages 1127–1137, Beijing, China. Association for Computational Linguistics.