

Capturing Argument Interaction in Semantic Role Labeling with Capsule Networks

Xinchi Chen¹ Chunchuan Lyu¹ Ivan Titov^{1,2}

xchen13@exseed.ed.ac.uk chunchuan.lv@gmail.com ititov@inf.ed.ac.uk

¹ ILCC, School of Informatics, University of Edinburgh

²ILLC, University of Amsterdam

Abstract

Semantic role labeling (SRL) involves extracting propositions (i.e. predicates and their typed arguments) from natural language sentences. State-of-the-art SRL models rely on powerful encoders (e.g., LSTMs) and do not model non-local interaction between arguments. We propose a new approach to modeling these interactions while maintaining efficient inference. Specifically, we use Capsule Networks (Sabour et al., 2017): each proposition is encoded as a tuple of *capsules*, one capsule per argument type (i.e. role). These tuples serve as embeddings of entire propositions. In every network layer, the capsules interact with each other and with representations of words in the sentence. Each iteration results in updated proposition embeddings and updated predictions about the SRL structure. Our model substantially outperforms the non-refinement baseline model on all 7 CoNLL-2019 languages and achieves state-of-the-art results on 5 languages (including English) for dependency SRL. We analyze the types of mistakes corrected by the refinement procedure. For example, each role is typically (but not always) filled with at most one argument. Whereas enforcing this approximate constraint is not useful with the modern SRL system, iterative procedure corrects the mistakes by capturing this intuition in a flexible and context-sensitive way.¹

1 Introduction

The task of semantic role labeling (SRL) involves the prediction of predicate-argument structure, i.e., both the identification of arguments and labeling them with underlying semantic roles. The shallow semantic structures have been shown beneficial in many natural language processing (NLP) applications, including information extrac-

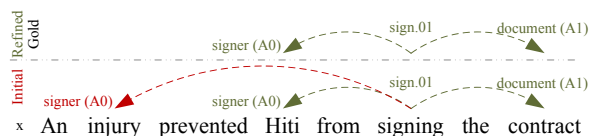


Figure 1: An example predicate-argument structure: green and red arcs denote correct and wrong predictions, respectively.

tion (Christensen et al., 2011), question answering (Eckert and Neves, 2018) and machine translation (Marcheggiani et al., 2018).

In this work, we focus on the dependency version of the SRL task (Surdeanu et al., 2008). An example of a dependency semantic-role structure is shown in Figure 1. Edges in the graph are marked with semantic roles, whereas predicates (typically verbs or nominalized verbs) are annotated with their senses.

Intuitively, there are many restrictions on potential predicate-argument structures. Consider the ‘role uniqueness’ constraint: each role is typically, but not always, realized at most once. For example, predicates have at most one *agent*. Similarly, depending on a verb class, only certain subcategorization patterns are licensed. Nevertheless, rather than modeling the interaction between argument labeling decisions, state-of-the-art semantic role labeling models (Marcheggiani and Titov, 2017; Cai et al., 2018; Li et al., 2019b) rely on powerful sentence encoders (e.g., multi-layer Bi-LSTMs (Zhou and Xu, 2015; Qian et al., 2017; Tan et al., 2018)). This contrasts with earlier work on SRL, which hinged on global declarative constraints on the labeling decisions (FitzGerald et al., 2015; Täckström et al., 2015; Das et al., 2012). Modern SRL systems are much more accurate and hence enforcing such hard and often approximate constraints is unlikely to be as beneficial (see our experiments in Section 7.2).

¹Code: <https://github.com/DalstonChen/CapNetSRL>.

Instead of using hard constraints, we propose to use a simple iterative structure-refinement procedure. It starts with independent predictions and refines them in every subsequent iteration. When refining a role prediction for a given candidate argument, information about the assignment of roles to other arguments gets integrated. Our intuition is that modeling interactions through the output spaces rather than hoping that the encoder somehow learns to capture them, provides a useful inductive bias to the model. In other words, capturing this interaction explicitly should mitigate overfitting. This may be especially useful in a lower-resource setting but, as we will see in experiments, it appears beneficial even in a high-resource setting.

We think of semantic role labeling as extracting propositions, i.e. predicates and argument-role tuples. In our example, the proposition is `sign.01(Arg0:signer = Hiti, Arg1:document = contract)`. Across the iterations, we maintain and refine not only predictions about propositions but also their embeddings. Each ‘slot’ (e.g., correspond to role `Arg0:signer`) is represented with a vector, encoding information about arguments assumed to be filling this slot. For example, if *Hiti* is predicted to fill slot `Arg0` in the first iteration then the slot representation will be computed based on the contextualized representation of that word and hence reflect this information. The combination of all slot embeddings (one per semantic role) constitutes the embedding of the proposition. The proposition embedding, along with the prediction about the semantic role structure, gets refined in every iteration of the algorithm.

Note that, in practice, the predictions in every iteration are soft, and hence proposition embeddings will encode current beliefs about the predicate-argument structure. The distributed representation of propositions provides an alternative (“dense-embedding”) view on the currently considered semantic structure, i.e. information extracted from the sentence. Intuitively, this representation can be readily tested to see how well current predictions satisfy selection restrictions (e.g., *contract* is a very natural filler for `Arg1:document`) and check if the arguments are compatible.

To get an intuition how the refinement mechanism may work, imagine that both *Hiti* and *injury*

are labeled as `Arg0` for `sign.01` in the first iteration. Hence, the representation of the slot `Arg0` will encode information about both predicted arguments. At the second iteration, the word *injury* will be aware that there is a much better candidate for filling the `signer` role, and the probability of assigning *injury* to this role will drop. As we will see in our experiments, whereas enforcing the hard uniqueness constraint is not useful, our iterative procedure corrects the mistakes of the base model by capturing interrelations between arguments in a flexible and context-sensitive way.

In order to operationalize the above idea, we take inspiration from the Capsule Networks (CNs) (Sabour et al., 2017). Note that we are not simply replacing Bi-LSTMs with generic CNs. Instead, we use CN to encode the structure of the refinement procedure sketched above. Each slot embedding is now a *capsule* and each proposition embedding is a tuple of capsules, one capsule per role. In every iteration (i.e. CN network layer), the capsules interact with each other and with representations of words in the sentence.

We experiment with our model on standard benchmarks for 7 languages from CoNLL-2019. Compared with the non-refinement baseline model, we observe substantial improvements from using the iterative procedure. The model achieves state-of-the-art performance in 5 languages, including English.

2 Base Dependency SRL Model

In dependency SRL, for each predicate p of a given sentence $x = \{x_1, x_2, \dots, x_n\}$ with n words, the model needs to predict roles $y = \{y_1, y_2, \dots, y_n\}$ for every word. The role can be *none*, signifying that the corresponding word is not an argument of the predicate.

We start with describing the factorized baseline model which is similar to that of Marcheggiani et al. (2017). It consists of three components: (1) an embedding layer; (2) an encoding layer and (3) an inference layer.

2.1 Embedding Layer

The first step is to map symbolic sentence x and predicate p into continuous embedded space:

$$\mathbf{e}_i = \text{Lookup}(x_i), \quad (1)$$

$$\mathbf{p} = \text{Lookup}(p), \quad (2)$$

where $\mathbf{e}_i \in \mathcal{R}^{d_e}$ and $\mathbf{p} \in \mathcal{R}^{d_e}$.

2.2 Encoding Layer

The encoding layer extracts features from input sentence x and predicate p . We extract features from input sentence x using stacked bidirectional LSTMs (Hochreiter and Schmidhuber, 1997):

$$\mathbf{x}_i = \text{Bi-LSTMs}(\mathbf{e}_i), \quad (3)$$

where $\mathbf{x}_i \in \mathcal{R}^{d_l}$. Then we represent each role logit of each word by a bi-linear operation with the target predicate:

$$b_{j|i} = \mathbf{x}_i \mathbf{W} \mathbf{p}, \quad (4)$$

where $\mathbf{W} \in \mathcal{R}^{d_l \times d_e}$ is a trainable parameter and $b_{j|i} \in \mathcal{R}$ is a scalar representing the logit of role j for word x_i .

2.3 Inference Layer

The probability $P(y|x, p)$ is then computed as

$$P(y|x, p) = \prod_i P(y_i|x, i, p), \quad (5)$$

$$= \prod_i \text{softmax}(\mathbf{b}_{\cdot|i})_{y_i}, \quad (6)$$

where $\mathbf{b}_{\cdot|i} = \{b_{1|i}, b_{2|i}, \dots, b_{|\mathcal{T}||i}\}$ and $|\mathcal{T}|$ denotes the number of role types.

3 Dependency based Semantic Role Labeling using Capsule Networks

Inspired by capsule networks, we use capsule structure for each role state to maintain information across iterations and employ the dynamic routing mechanism to derive the role logits $b_{j|i}$ iteratively. Figure 2 illustrates the architecture of the proposed model.

3.1 Capsule Structure

We start by introducing two capsule layers: (1) the word capsule layer and (2) the role capsule layer.

3.1.1 Word Capsule Layer

The word capsule layer is comprised of capsules representing the roles of each word. Given sentence representation \mathbf{x}_i and predicate embedding \mathbf{p} , the word capsule layer is derived as:

$$\mathbf{u}_{j|i}^k = \mathbf{x}_i \mathbf{W}_j^k \mathbf{p}; \quad k = 1, 2, \dots, K, \quad (7)$$

where $\mathbf{W}_j^k \in \mathcal{R}^{d_l \times d_e}$ and $\mathbf{u}_{j|i} \in \mathcal{R}^K$ is the capsule vector for role j of word x_i . K denotes the capsule size. Intuitively, the capsule encodes the argument-specific information relevant to deciding if role j is suitable for the word. These capsules do not get iteratively updated.

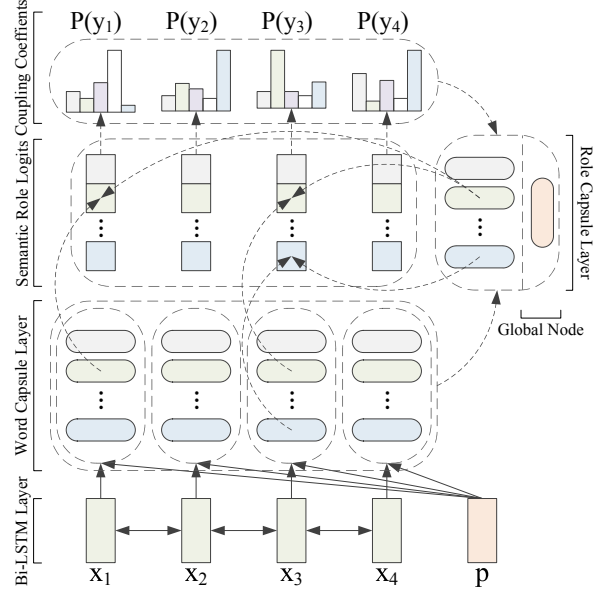


Figure 2: Architecture of the proposed CapsuleNet with the global capsule node. The dashed arcs are iterable.

3.1.2 Role Capsule Layer

As discussed in the introduction, the role capsule layer could be viewed as an embedding of a proposition. The capsule network generates the capsules in the layer using “routing-by-agreement”. This process can be regarded as a pooling operation. Capsules in the role capsule layer at t -th iteration are derived as:

$$\mathbf{v}_j^{(t)} = \text{Squash}(\mathbf{s}_j^{(t)}) = \frac{\|\mathbf{s}_j^{(t)}\|^2}{1 + \|\mathbf{s}_j^{(t)}\|^2} \frac{\mathbf{s}_j^{(t)}}{\|\mathbf{s}_j^{(t)}\|}, \quad (8)$$

where intermediate capsule $\mathbf{s}_j^{(t)}$ is generated with the linear combination of capsules in the word capsule layer with weights $c_{ij}^{(t)}$:

$$\mathbf{s}_j^{(t)} = \sum_i c_{ij}^{(t)} \mathbf{u}_{j|i}. \quad (9)$$

Here, $c_{ij}^{(t)}$ are coupling coefficients, calculated by “softmax” function over role logits $b_{j|i}^{(t)}$. The coefficients $c_{ij}^{(t)}$ can be interpreted as the probability that word x_i is assigned role j :

$$\mathbf{c}_i^{(t)} = \text{softmax}(\mathbf{b}_{\cdot|i}^{(t)}). \quad (10)$$

The role logits $b_{j|i}^{(t)}$ are decided by the iterative *dynamic routing* process. The Squash operation will deactivate capsules receiving small input $\mathbf{s}_j^{(t)}$ (i.e. roles not predicted in the sentence) by pushing them further to the $\mathbf{0}$ vector.

Input: # of iterations T , all capsules in l_w $\mathbf{u}_{j|i}$

Result: semantic role logits $b_{j|i}^{(T)}$

Initialization: $b_{j|i}^{(0)} = 0$;

for $t \in [0, 1, \dots, T - 1]$ **do**

 for all capsules in l_w : $\mathbf{c}_i^{(t)} = \text{softmax}(\mathbf{b}_{\cdot|i}^{(t)})$;

 for all capsules in l_r : $\mathbf{s}_j^{(t)} = \sum_i c_{ij}^{(t)} \mathbf{u}_{j|i}$;

 for all capsules in l_r : $\mathbf{v}_j^{(t)} = \text{Squash}(\mathbf{s}_j^{(t)})$;

 for all capsules in l_w and l_r : $b_{j|i}^{(t+1)} = b_{j|i}^{(t)} + \mathbf{v}_j^{(t)} \mathbf{W} \mathbf{u}_{j|i}$;

end

Algorithm 1: Dynamic routing algorithm. l_w and l_r denote word capsule layer and role capsule layer, respectively.

3.2 Dynamic Routing

The dynamic routing process involves T iterations. The role logits $b_{j|i}$ before first iteration are all set to zeros: $b_{j|i}^{(0)} = 0$. Then, the dynamic routing process updates the role logits $b_{j|i}$ by modeling agreement between capsules in two layers:

$$b_{j|i}^{(t+1)} = b_{j|i}^{(t)} + \mathbf{v}_j^{(t)} \mathbf{W} \mathbf{u}_{j|i}, \quad (11)$$

where $\mathbf{W} \in \mathcal{R}^{K \times K}$.

The whole dynamic routing process is shown in Algorithm 1. The dynamic routing process can be regarded as the role refinement procedure (see Section 5 for details).

4 Incorporating Global Information

When computing the j -th role capsule representation (Eq 9), the information originating from an i -th word (i.e. $\mathbf{u}_{j|i}$) is weighted by the probability of assigning role j to word x_i (i.e. c_{ij}). In other words, the role capsule receives messages only from words assigned to its role. This implies that the only interaction the capsule network can model is competition between arguments for a role.² Note though that this is different from imposing the hard role-uniqueness constraint, as the network does this dynamically in a specific context. Still, this is a strong restriction.

In order to make the model more expressive, we further introduce a global node $\mathbf{g}^{(t)}$ to incorporate global information about all arguments at the current iterations. The global node is a compressed representation of the entire proposition, and used in the prediction of all arguments, thus permitting arbitrary interaction across arguments. The global

²In principle, it can model the opposite, i.e. collaboration / attraction but it is unlikely to be useful in SRL.

node $\mathbf{g}^{(t)}$ at t -th iteration is derived as:

$$\mathbf{g}^{(t)} = \mathbf{W} \mathbf{s}^{(t)}, \quad (12)$$

where $\mathbf{W} \in \mathcal{R}^{K \times (K \cdot |\mathcal{T}|)}$. Here, $\mathbf{s}^{(t)} = \mathbf{s}_1^{(t)} \oplus \mathbf{s}_2^{(t)} \oplus \dots \oplus \mathbf{s}_{|\mathcal{T}|}^{(t)} \in \mathcal{R}^{K \cdot |\mathcal{T}|}$ is the concatenation of all capsules in the role capsule layer.

We append an additional term for the role logits update in Eq (11):

$$b_{j|i}^{(t+1)} = b_{j|i}^{(t)} + \mathbf{v}_j^{(t)} \mathbf{W} \mathbf{u}_{j|i} + \mathbf{g}^{(t)} \mathbf{W}_g \mathbf{u}_{j|i}, \quad (13)$$

where $\mathbf{W} \in \mathcal{R}^{K \times K}$ and $\mathbf{W}_g \in \mathcal{R}^{K \times K}$.

5 Refinement

The dynamic routing process can be seen as iterative role refinement. Concretely, the coupling coefficients $\mathbf{c}_i^{(t)}$ in Eq (10) can be interpreted as the predicted distribution of the semantic roles for word x_i in Eq (5) at t -th iteration:

$$\mathbf{c}_i^{(t)} = P^{(t)}(\cdot | x, i, p). \quad (14)$$

Since dynamic routing is an iterative process, semantic role distribution $\mathbf{c}_i^{(t)}$ in iteration t will affect the semantic role distribution $\mathbf{c}_i^{(t+1)}$ in next iteration $t + 1$:

$$\mathbf{c}_i^{(t+1)} = f(\mathbf{c}_i^{(t)}, x, i, p), \quad (15)$$

where $f(\cdot)$ denotes the refinement function defined by the operations in each dynamic routing iteration.

6 Training

We minimize the following loss function $\mathcal{L}(\theta)$:

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_i \log P^{(T)}(y_i | x, i, p; \theta) + \lambda \|\theta\|_2^2, \quad (16)$$

where λ is a hyper-parameter for the regularization term and $P^{(T)}(y_i | x, i, p; \theta) = \mathbf{c}_i^{(T)}$. Unlike standard refinement methods (Belanger et al., 2017; Lee et al., 2018) which sum losses across all refinement iterations, our loss is only based on the prediction made at the last iteration. This encourages the model to rely on the refinement process rather than do it in one shot.

Our baseline model is trained analogously, but using the cross-entropy for the independent classifiers (Eq (5)).

Uniqueness Constraint Assumption As we discussed earlier, for a given target predicate, each semantic role will typically appear at most once. To encode this intuition, we propose another loss term $\mathcal{L}_u(\theta)$:

$$\mathcal{L}_u(\theta) = \frac{1}{|\mathcal{T}|} \sum_j \log \text{softmax}(\mathbf{b}_{j|\cdot}^{(T)}), \quad (17)$$

where $\mathbf{b}_{j|\cdot}^{(T)}$ are the semantic role logits in T -th iteration. Thus, the final loss $\mathcal{L}^*(\theta)$ is the combination of the two losses:

$$\mathcal{L}^*(\theta) = \mathcal{L}(\theta) + \eta \mathcal{L}_u(\theta), \quad (18)$$

where η is a discount coefficient.

7 Experiments

7.1 Datasets & Configuration

we conduct experiments on CoNLL-2009 (Hajič et al., 2009) for all languages, including Catalan (Ca), Chinese (Zh), Czech (Cz), English (En), German (De), Japanese (Jp) and Spanish (Es). As standard we do not consider predicate identification: the predicate position is provided as an input feature. We use the labeled ‘semantic’ F1 which jointly scores sense prediction and role labeling.

We use ELMo (Peters et al., 2018) (dimension d_e as 1024) for English, and FastText embeddings (Grave et al., 2018) (dimension d_e as 300) for all other languages on both the baseline model and the proposed CapsuleNet. LSTM state dimension d_l is 500. Capsule size K is 16. Batch size is 32. The coefficient for the regularization term λ is 0.0004. We employ Adam (Kingma and Ba, 2015) as the optimizer and the initial learning rate α is set to 0.0001. Syntactic information is not utilized in our model.

7.2 Model Selection

Table 1 shows the performance of our model trained with loss $\mathcal{L}^*(\theta)$ for different values of discount coefficient η on the English development set. The model achieves the best performance when η equals to 0. It implies that adding uniqueness constraint on loss actually hurts the performance. Thus, we use the loss $\mathcal{L}^*(\theta)$ with η equals to 0 in the rest of the experiments, which is equivalent to the loss $\mathcal{L}(\theta)$ in Eq (16). We also observe that the model with 2 refinement iterations performs the best (89.92% F1).

η	# of Iters	P	R	F	EM
1.0	1	89.31	88.31	88.80	63.21
	2	88.34	88.92	88.63	62.80
	3	88.92	88.97	88.94	63.51
0.1	1	89.16	87.99	88.57	62.94
	2	88.70	88.89	88.80	63.00
	3	89.75	89.40	89.57	66.01
0.0	1	89.59	90.06	89.82	66.74
	2	89.88	89.97	89.92	66.93
	3	89.25	89.61	89.43	65.98

Table 1: The performance of our model (w/o global node) with different discount coefficients η of loss $\mathcal{L}^*(\theta)$ on the English development set. EM denotes the ratio of exact match on propositions.

Models	Test	Ood
Lei et al. (2015)	86.60	75.60
FitzGerald et al. (2015)	87.70	75.50
Foland and Martin (2015)	86.00	75.90
Roth and Lapata (2016)	87.90	76.50
Swayamdipta et al. (2016)	85.00	-
Marcheggiani et al. (2017)	87.70	77.70
Marcheggiani and Titov (2017)	89.10	78.90
He et al. (2018)	89.50	79.30
Li et al. (2018)	89.80	79.80
Cai et al. (2018)	89.60	79.00
Mulcaire et al. (2018)	87.24	-
Li et al. (2019b) [†]	90.40	81.50
Baseline [†]	90.49	81.95
CapsuleNet[†] (This Work)	91.06	82.72

Table 2: The F1 scores of previous systems on the English test set and out-of-domain (Ood) set. [†] denotes that the model uses Elmo.

7.3 Overall Results

Table 2 compares our model with previous state-of-the-art SRL systems on English. Some of the systems (Lei et al., 2015; Li et al., 2018; Cai et al., 2018) only use local features, whereas others (Swayamdipta et al., 2016) incorporate global information at the expense of greedy decoding. Additionally, a number of systems exploit syntactic information (Roth and Lapata, 2016; Marcheggiani and Titov, 2017; He et al., 2018). Some of the results are obtained with ensemble systems (FitzGerald et al., 2015; Roth and Lapata, 2016).

As we observe, the baseline model (see Section 2) is quite strong, and outperforms the previous state-of-the-art systems on both in-domain and out-of-domain sets on English. The proposed CapsuleNet outperforms the baseline model on English (e.g. obtaining 91.06% F1 on the English test set), which shows the effectiveness of the cap-

Models	Development				Test				Ood			
	P	R	F	EM	P	R	F	EM	P	R	F	EM
CapsuleNet SRL	89.63	90.25	89.94	67.15	90.74	91.38	91.06	69.90	82.66	82.78	82.72	53.14
w/o Global Node	89.88	89.97	89.92	66.93	90.95	91.15	91.05	69.65	82.45	82.27	82.36	51.95
w/o Role Capsule Layer	89.95	89.64	89.80	66.70	90.82	90.66	90.74	68.46	82.26	81.64	81.95	50.75
w/o Word Capsule Layer (Baseline)	89.47	89.61	89.54	66.17	90.31	90.68	90.49	68.27	82.16	81.74	81.95	50.75

Table 3: Ablation results in English development and test sets. EM denotes the ratio of the exact match on propositions.

sule network framework. The improvement on the out-of-domain set implies that our model is robust to domain variations.

7.4 Ablation

Table 3 gives the performance of models with ablation on some key components, which shows the contribution of each component in our model.

- The model **without the global node** is described in Section 3.
- The model that further **removes the role capsule layer** takes the mean of capsules $\mathbf{u}_{j|i}$ in Eq (7) of word capsule layer as the semantic role logits $b_{j|i}$:

$$b_{j|i} = \frac{1}{K} \sum_k^K \mathbf{u}_{j|i}^k, \quad (19)$$

where K denotes the capsule size.

- The model that additionally **removes the word capsule layer** is exactly equivalent to the baseline model described in Section 2.

As we observe, CapsuleNet with all components performs the best on both development and test sets on English. The model without using the global node performs well too. It obtains 91.05% F1 on the English test set, almost the same performance as full CapsuleNet. But on the English out-of-domain set, without the global node, the performance drops from 82.72% F1 to 82.36% F1. It implies that the global node helps in model generalization. Further, once the role capsule layer is removed, the performance drops sharply. Note that the model without the role capsule layer does not use refinements and hence does not model argument interaction. It takes the mean of capsules $\mathbf{u}_{j|i}$ in the word capsule layer as the semantic role logits $b_{j|i}$ (see Eq (19)), and hence could be viewed as an enhanced (‘ensembled’) version of the baseline

Sent Len	# of Props	# of Args	# of Props
0 - 9	181	1	2,591
10 - 19	1,806	2	4,497
20 - 29	3,514	3	2,189
30 - 39	3,102	4	889
40 - 49	1,383	5	259
50 - 59	391	6	39
60 - 69	121	7	7

Table 4: Numbers of propositions with different sentence lengths and different numbers of arguments on the English test set.

model. Note that we only introduced a very limited number of parameters for the dynamic routing mechanism (see Eq (11-13)). This suggests that the dynamic routing mechanism does genuinely capture interactions between argument labeling decisions and the performance benefits from the refinement process.

7.5 Error Analysis

The performance of our model while varying the sentence length and the number of arguments per proposition is shown in Figure 3. The statistics of the subsets are in Table 4. Our model consistently outperforms the baseline model, except on sentences of between 50 and 59 words. Note that the subset is small (only 391 sentences), so the effect may be random.

7.6 Effect of Refinement

7.6.1 Refinement Performance

We take a CapsuleNet trained to perform refinements in two iterations and vary the number of iterations at test time (Figure 4). CapsuleNets with and without the global node perform very differently. CapsuleNet without the global node reduces the recall and increases the precision with every iteration. This corresponds to our expectations: the version of the network is only capable of modeling competition and hence continues with this filtering process. In contrast, CapsuleNet with the global

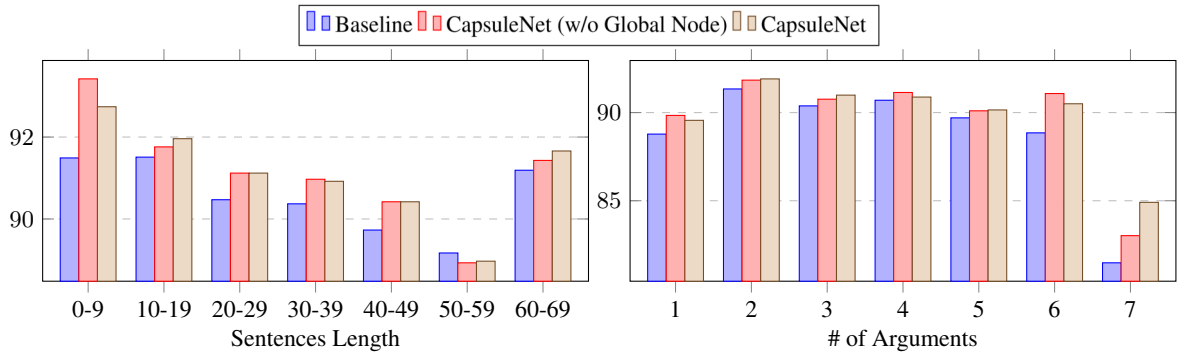


Figure 3: The F1 scores of different models while varying sentence length and argument number on the English test set.

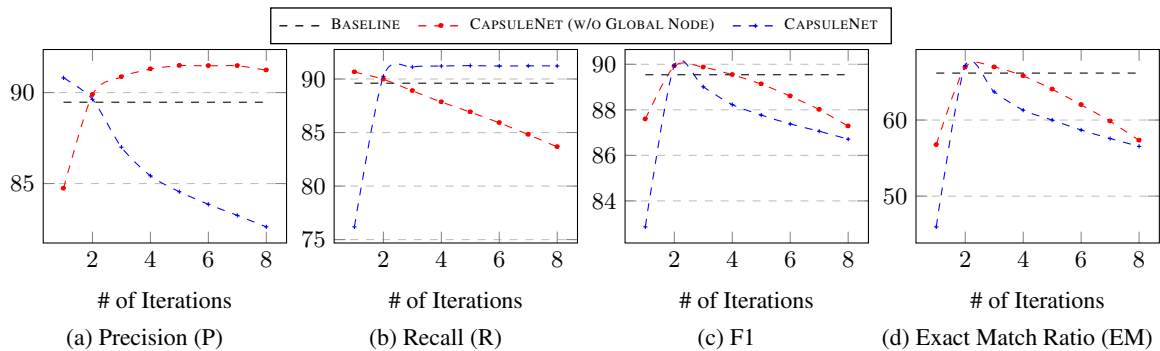


Figure 4: The performance of different models while varying the number of refinement iterations on the English development set.

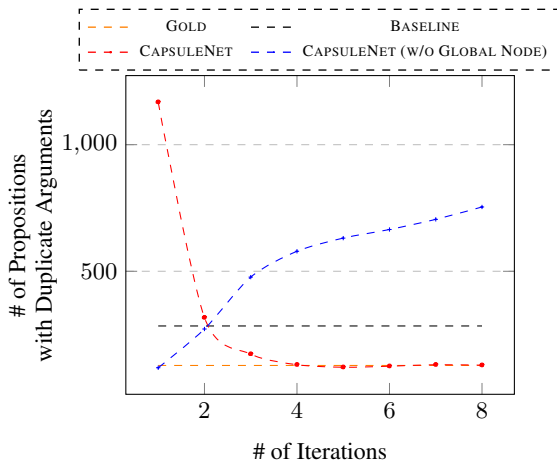


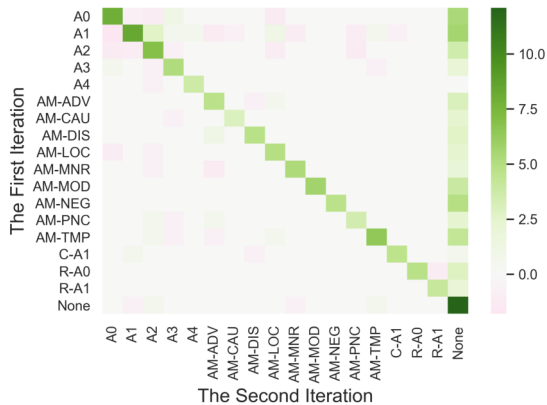
Figure 5: The number of propositions with duplicate arguments for various numbers of refinement iterations on the English development set.

node reduces the precision as the number of iteration grows. This model is less transparent, so it harder to see why it chooses this refinement strategy. As expected, the F1 scores for both models peak at the second iteration, the iteration number used in the training phase. The trend for the exact match score is consistent with the F1 score.

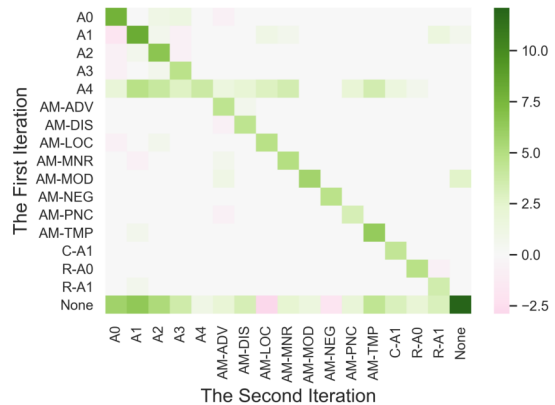
7.6.2 Duplicate Arguments

We measure the degree to which the role uniqueness constraint is violated by both models. We plot the number of violations as the function of the number of iterations (Figure 5). Recall that the violations do not imply errors in predictions as even the gold-standard data does not always satisfy the constraint (see the orange line in the figure). As expected, CapsuleNet without the global node captures competition and focuses on enforcing this constraint. Interestingly, it converges to keeping the same fraction of violations as the one observed in the gold standard data. In contrast, the full model increasingly ignores the constraint in later iterations. This is consistent with the over-generation trend evident from the precision and recall plots discussed above.

Figure 6 illustrates how many roles get changed between consecutive iterations of CapsuleNet, with and without the global node. Green indicates how many correct changes have been made, while red shows how many errors have been introduced. Since the number of changes is very large, we represent the non-zero number q in the



(a) Capsule Network w/o Global Node



(b) Capsule Network with Global Node

Figure 6: Changes in labeling between consecutive iterations on the English development set. Only argument types appearing more than 50 times are listed. “None” type denotes “NOT an argument”. Green and red nodes denote the numbers of correct and wrong role transitions have been made respectively. The numbers are transformed into log space.

Model	Ja	Es	Ca	De	Cz	Zh	En	Avg.
Previous Best Single Model	78.69	80.50	80.32	80.10	86.02	84.30	90.40	82.90
Baseline Model	80.12	81.00	81.39	76.01	87.79	81.05	90.49	82.55
CapsuleNet SRL (This Work)	81.26	81.32	81.65	76.44	88.08	81.65	91.06	83.07

Table 5: Labeled F1 score (including senses) for all languages on the CoNLL-2009 in-domain test sets. For previous best result, Japanese (Ja) is from [Watanabe et al. \(2010\)](#), Catalan (Ca) is from [Zhao et al. \(2009\)](#), Spanish (Es) and German (De) are from [Roth and Lapata \(2016\)](#), Czech (Cz) is from [Henderson et al. \(2013\)](#), Chinese (Zh) is from [Cai et al. \(2018\)](#) and English (En) is from [Li et al. \(2019b\)](#).

log space $\tilde{q} = \text{sign}(q) \log(|q|)$.

As we expected, for both models, the majority of changes are correct, leading to better overall performance. We can again see the same trends. CapsuleNet without the global node tends to filter our arguments by changing them to “None”. The reverse is true for the full model.

7.7 Multilingual Results

Table 5 gives the results of the proposed CapsuleNet SRL (with global node) on the in-domain test sets of all languages from CoNLL-2009. As shown in Table 5, the proposed model consistently outperforms the non-refinement baseline model and achieves state-of-the-art performance on Catalan (Ca), Czech (Cz), English (En), Japanese (Jp) and Spanish (Es). Interestingly, the effectiveness of the refinement method does not seem to be dependent on the dataset size: the improvements on the smallest (Japanese) and the largest datasets (English) are among the largest.

8 Additional Related Work

The capsule networks have been recently applied to a number of NLP tasks ([Xiao et al., 2018](#); [Gong et al., 2018](#)). In particular, [Yang et al. \(2018\)](#) represent text classification labels by a layer of capsules, and take the capsule actions as the classification probability. Using a similar method, [Xia et al. \(2018\)](#) perform intent detection with the capsule networks. [Wang et al. \(2018\)](#) and [Li et al. \(2019a\)](#) use capsule networks to capture rich features for machine translation. More closely to our work, [Zhang et al. \(2018\)](#) and [Zhang et al. \(2019\)](#) adopt the capsule networks for relation extraction. The previous models apply the capsule networks to problems that have a fixed number of components in the output. Their approach cannot be directly applied to our setting.

9 Conclusions & Future Work

State-of-the-art dependency SRL methods do not account for any interaction between role labeling decisions. We propose an iterative approach to

SRL. In each iteration, we refine both predictions of the semantic structure (i.e. a discrete structure) and also a distributed representation of the proposition (i.e. the predicate and its predicted arguments). The iterative refinement process lets the model capture interactions between the decisions. We relied on the capsule networks to operationalize this intuition. We demonstrate that our model is effective, and results in improvements over a strong factorized baseline and state-of-the-art results on standard benchmarks for 5 languages (Catalan, Czech, English, Japanese and Spanish) from CoNLL-2019. In future work, we would like to extend the approach to modeling interaction between multiple predicate-argument structures in a sentence as well as to other semantic formalisms (e.g., abstract meaning representations (Banarescu et al., 2013)).

Acknowledgments

We thank Diego Marcheggiani, Jonathan Mallinson and Philip Williams for constructive feedback and suggestions, as well as anonymous reviewers for their comments. The project was supported by the European Research Council (ERC StG BroadSem 678254) and the Dutch National Science Foundation (NWO VIDI 639.022.518).

References

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.
- David Belanger, Bishan Yang, and Andrew McCallum. 2017. [End-to-end learning for structured prediction energy networks](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 429–439.
- Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntax-agnostic over syntax-aware? *arXiv preprint arXiv:1808.03815*.
- Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. [An analysis of open information extraction based on semantic role labeling](#). In *Proceedings of the Sixth International Conference on Knowledge Capture, K-CAP '11*, pages 113–120, New York, NY, USA. ACM.
- Dipanjan Das, André FT Martins, and Noah A Smith. 2012. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 209–217. Association for Computational Linguistics.
- Fabian Eckert and Mariana Neves. 2018. [Semantic role labeling tools for biomedical question answering: a study of selected tools on the BioASQ datasets](#). In *Proceedings of the 6th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.
- Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970.
- William Foland and James Martin. 2015. Dependency-based semantic role labeling using convolutional neural networks. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 279–288.
- Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuanjing Huang. 2018. Information aggregation via dynamic routing for sequence encoding. *arXiv preprint arXiv:1806.01501*.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning word vectors for 157 languages](#). In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.
- 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: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. 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.
- James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. [Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model](#). *Computational Linguistics*, 39(4):949–998.

- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Comput.*, 9(8):1735–1780.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. [Deterministic non-autoregressive neural sequence modeling by iterative refinement](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.
- Tao Lei, Yuan Zhang, Lluís Marquez, Alessandro Moschitti, and Regina Barzilay. 2015. 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*, pages 1150–1160.
- Jian Li, Baosong Yang, Zi-Yi Dou, Xing Wang, Michael R Lyu, and Zhaopeng Tu. 2019a. Information aggregation for multi-head attention with routing-by-agreement. *arXiv preprint arXiv:1904.03100*.
- Zuchao Li, Shexia He, Jiayun 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.
- Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019b. Dependency or span, end-to-end uniform semantic role labeling. *arXiv preprint arXiv:1901.05280*.
- Diego Marcheggiani, Joost Bastings, and Ivan Titov. 2018. [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)*, pages 486–492, New Orleans, Louisiana. Association for Computational Linguistics.
- Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. *arXiv preprint arXiv:1701.02593*.
- Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.
- Phoebe Mulcaire, Swabha Swayamdipta, and Noah Smith. 2018. Polyglot semantic role labeling. *arXiv preprint arXiv:1805.11598*.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Feng Qian, Lei Sha, Baobao Chang, LuChen Liu, and Ming Zhang. 2017. [Syntax aware LSTM model for semantic role labeling](#). In *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pages 27–32, Copenhagen, Denmark. Association for Computational Linguistics.
- Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. *arXiv preprint arXiv:1605.07515*.
- Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. [Dynamic routing between capsules](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3856–3866. Curran Associates, Inc.
- 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 *Proceedings of the Twelfth Conference on Computational Natural Language Learning, CoNLL '08*, pages 159–177, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Greedy, joint syntactic-semantic parsing with stack lstms. *arXiv preprint arXiv:1606.08954*.
- 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 *AAAI*.
- Mingxuan Wang, Jun Xie, Zhixing Tan, Jinsong Su, et al. 2018. Towards linear time neural machine translation with capsule networks. *arXiv preprint arXiv:1811.00287*.
- Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. 2010. [A structured model for joint learning of argument roles and predicate senses](#). In *Proceedings of the ACL 2010 Conference Short Papers*, pages 98–102, Uppsala, Sweden. Association for Computational Linguistics.
- Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. [Zero-shot user intent detection via capsule neural networks](#). In

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3090–3099, Brussels, Belgium. Association for Computational Linguistics.

Liqiang Xiao, Honglun Zhang, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. [MCapsNet: Capsule network for text with multi-task learning](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4565–4574, Brussels, Belgium. Association for Computational Linguistics.

Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. [Investigating capsule networks with dynamic routing for text classification](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3110–3119, Brussels, Belgium. Association for Computational Linguistics.

Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018. [Attention-based capsule networks with dynamic routing for relation extraction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 986–992, Brussels, Belgium. Association for Computational Linguistics.

Xinsong Zhang, Pengshuai Li, Weimin Jia, and Hai Zhao. 2019. Multi-labeled relation extraction with attentive capsule network. *CoRR*, abs/1811.04354.

Hai Zhao, Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Multilingual dependency learning: Exploiting rich features for tagging syntactic and semantic dependencies. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 61–66. 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.