KU Leuven - CSAI

Exercise Session 3
Semantic role labeling

For questions contact:
Toledo Forum for Questions (under Discussions) or nlp@ls.kuleuven.be

Quyhn Do Thi, Victor Milewski and Marie-Francine Moens

1 Semantic Role Labeling with CRF

In this exercise, we work on a sub-task of semantic role labeling: argument classification. Given a certain sentence and the position of the predicate in that sentence, our goal is to build a linear-chain conditional random field (CRF) model that assigns the correct semantic role label to each argument of the predicate. In this exercise we limit ourself to 3 different labels:

$$A0 = \text{agent},\ A1 = \text{patient},\ \text{AM-MNR} = \text{manner}.$$

Some toy SRL data could be:

He_{A0} $\text{loves}_{\text{predicate}}$ dogs_{A1} $\text{tremendously}_{\text{AM-MNR}}$

He_{A0} $\text{ran}_{\text{predicate}}$ $\text{fast}_{\text{AM-MNR}}$

The CRF model $P(L|s, j)$ gives the probability for a labeling sequence L (one label per argument) for sentence s with a predicate at position j. It uses K different learned weights $\lambda_k \in \mathbb{R}$ and feature functions f_k for $k \in \{1, \ldots, K\}$ where $f_k(\ldots) \in \{0, 1\}$ by first calculating a score:

$$score(L|s, j) = \sum_{i=1}^{C(s,j)} \sum_{k=1}^{K} \lambda_k f_k(s, i, j, l_i, l_{i-1})$$

$C(s, j)$ is the number of arguments to be labeled for a given predicate (e.g., for 'loves' in the first sentence, there are 3 arguments to be labeled: 'He', 'dogs' and 'tremendously', so for example

$$C(s = [\text{'He'}, \text{'loves'}, \text{'dogs'}, \text{'tremendously'}], j = 2) = 3$$

l_i is the label for the ith argument in the sentence. For the first sentence, L would be $[\text{A0}, \text{A1}, \text{AM-MNR}]$.

This score is then converted to a probability with a softmax:

$$P(L|s, j) = \frac{exp(score(L|s, j))}{\sum_{L'} exp(score(L'|s, j))}$$

Where L' ranges over all possible answers (including wrong ones).

Question 1.A: designing feature functions

Given the toy data provided above, design three simple feature sets for this task (you essentially have to make all combinations of the given variable types).

- (Argument word)-(role label) feature
- (Predicate word)-(role label) feature
- (Role label)-(role label) transition features (include a *Null* feature as first but not second)

You can use $a(s, j)$ to indicate the subset of sentence s that corresponds to the argument words for predicate j. For example, $a(s = [\text{'He'}, \text{'loves'}, \text{'dogs'}, \text{'tremendously'}], j = 2) = [\text{'He'}, \text{'dogs'}, \text{'tremendously'}]$. Note that $C(s, j)$ is the length of $a(s, j)$.
How many feature functions can you build for each set, using the two examples as training data? Assume that each role label can only occur once per predicate. Do you include feature functions that are never observed in the training data? What could be an advantage of doing this? What could be a disadvantage?

Question 1.B: training the model

To train the model, we employ stochastic gradient ascent on a maximum log-likelihood objective: $\mathcal{L}(\lambda) = log(\prod_{s,j} P(L|s,j)) = \sum_{s,j} logP(L|s,j)$
We do this by updating the parameters λ_k for each training example. The derivative w.r.t. a particular weight λ_q is

$$\frac{\partial}{\partial \lambda_q} \log P(L|s,j) = \sum_{i=1}^{C(s,j)} f_q(s,i,j,l_i,l_{i-1}) - \sum_{L'} P(L'|s,j) * \sum_{i=1}^{C(s,j)} f_q(s,i,j,l'_i,l'_{i-1})$$

The weight update for SGD with learning rate α is:

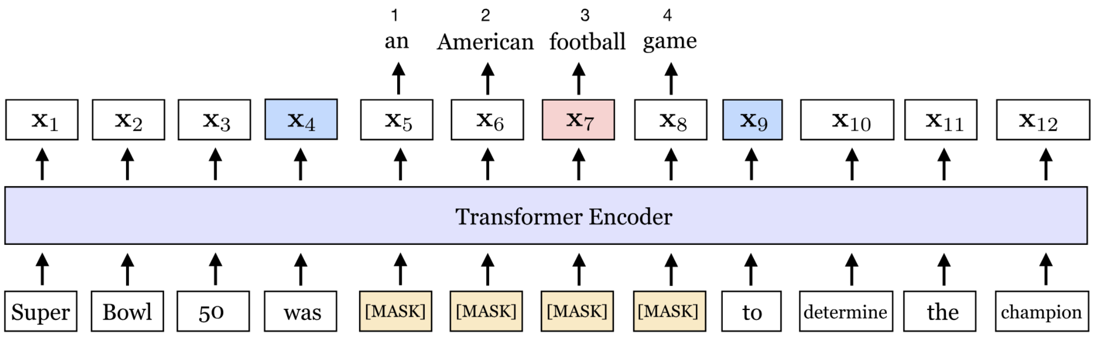
$$\lambda_q^{new} = \lambda_q + \alpha * \frac{\partial}{\partial \lambda_q} \log P(L|s,j)$$

Assuming that only the second example sentence is used for training and only features observed in the training data are used in the model, compute the weight vector λ for 2 iterations when using stochastic gradient descent to optimize the objective function (α = 0.1, and λ is set to zeros as initialization).

2 SpanBERT

In this question we will take a look at the SpanBERT model [1]. Figure 1 shows the architecture of the model and how it is pre-trained (in a self-supervised manner).

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



1 an
2 American
3 football
4 game
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12
Transformer Encoder
Super Bowl 50 was [MASK] [MASK] [MASK] [MASK] to determine the champion

Figure 1: An illustration of SpanBERT training. The span an *American football game* is masked. The SBO uses the output representations of the boundary tokens, $\boldsymbol{x}_4$ and $\boldsymbol{x}_9$ (in blue), to predict each token in the masked span. The equation shows the MLM and SBO loss terms for predicting the token, *football* (in pink), which as marked by the position embedding $\boldsymbol{p}_3$, is the third token from x_4.

3.A: Self-supervised pre-training with masked spans

The model trains by masking around 15% of the words in the sentence, just as with BERT. The difference is that BERT masks random words, in SpanBERT we only want to mask spans.

Can you think of two methods on how to choose spans of words to be masked during pre-training?

3.B: SpanBERT finetuned for downstream SRL

We want to use a pre-trained SpanBERT for a different downstream task, semantic role labeling. In this subquestion we will discuss if this would be a good approach to solving this task.
When the model receives a sentence and its predicate, the task of the model is to assign a label to each of the words in the sentence. The possible labels are:

```
[A0, A1, O]
```

1. Using your previous obtained knowledge about SRL, would it make sense to use SpanBERT? Do you think this would perform well?
2. Next we must decide how to train (finetune) the model. We decided that it would be best to train it using a multitask objective.
   First, the model follows the span boundary objective (SBO) function. We randomly mask spans from the input, and the model must predict the words for the masked span.
   Next, after filling the mask with the predictions, we use cross entropy for making the categorical SRL predictions.
   Write down the complete multitask objective function.

   Assume N sentences, where each sentence $X^n, n \in 1 \ldots N$ is a sequence of T^n words $x_1^n \ldots x_{T^n}^n$.
   Each sentence has a set of S^n masked spans. Each span is defined by its length K_i^n and start position c_i^n.
   For each word x_j^n, there is also a position embedding p_j^n, and an SRL label z_j^n.
   Finally, the model parameters are denoted with Θ. Θ_c are the common parameters involved in both objectives, Θ_{SBO} the parameters only involved in the SBO loss, and Θ_{CAT} those only involved in the CAT loss.

References

[1] Mandar Joshi et al. “Spanbert: Improving pre-training by representing and predicting spans”. In: *Transactions of the association for computational linguistics* 8 (2020), pp. 64–77.