1 Preliminaries: Constituent and Dependency Trees

Let $w = (w_1, \dots, w_L)$ be a sentence.

A *constituent tree* is a rooted tree whose leaves are the words $(w_i)_{i=1}^N$ and internal nodes are constituents.

A *constituent* is a triple $(Z, \mathcal{Y}, h)$ containing, respectively, its label, yield, and lexical head which satisfy some constraints[1].

A constituent is *discontinuous* if its yield is not contiguous.

A *dependency tree* is a rooted tree spanning the words in the sentence $(w_i)_{i=1}^N$. Each edge is labelled and connects a parent word (head) to a child word (dependency).

Fernández-González and Martins (2015) show that under quite general conditions[2], constituent trees are isomorphic to dependency trees in which the edges contain information about constituent labels and attachment order.

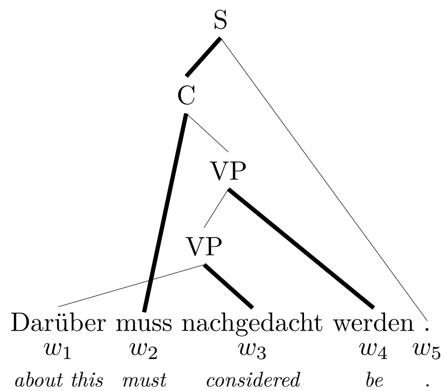


Figure 1.1: Example of a discontinuous constituent tree for the German sentence *'Darüber muss nachgedacht werden.'* ('this must be considered.'). Bold lines indicate head words.

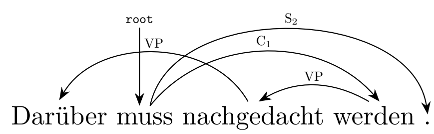
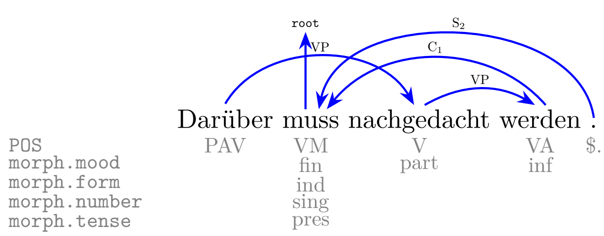


Figure 1.2: The dependency tree corresonding to the constituent tree in Fig. 1.1.

2 Mathematical Model



Given regressor $w = (w_i)_{i=1}^N$, our goal is to model its dependency tree along with information like its part-of-speech (POS) and morphology.

Our parser will work from the bottom-up, so we will think of *arcs* going from every child w_i to its parent w_i^{arc}.

Denote the regressand by $y = (y_i)_{i=1}^N$ where $y_i = \left(w_i^{\text{arc}}, w_i^{\text{lab}}, w_i^{\text{ord}}, w_i^{\text{pos}}, w_i^{\text{morph}}\right)$. Define $y_{<i} = (y_1, \dots, y_i)$ for each $i = 2, \dots, N$ and $y_{<1} = 0$.

[1] A leaf node can be modelled as a constituent whose yield contains only its head. For every constituent $(Z, \mathcal{Y}, h)$ with children $\{(A_k, \mathcal{X}_k, m_k)\}$, 1. $\mathcal{Y} = \bigcup_{k=1}^K \mathcal{X}_k$, and 2. there is a unique k such that $h = m_k$.

[2] The constituent trees must be unaryless.

Assumption. For each $i = 1, \ldots, N$, the random variables $w_i^{\text{lab}}, w_i^{\text{ord}}, w_i^{\text{pos}}$ and w_i^{morph} are mutually independent conditional on w_i^{arc}, w and $y_{<i}$.[3]

We can decompose the conditional probability of y given w:

$$\begin{aligned} p(y \mid w) &= \prod_{i=1}^{N} p(y_i \mid y_{<i}, w) && (2.1) \\ &= \boxed{\prod_{i=1}^{N} p(w_i^{\text{arc}} \mid y_{<i}, w) p(w_i^{\text{lab}} \mid w_i^{\text{arc}}, y_{<i}, w) p(w_i^{\text{ord}} \mid w_i^{\text{arc}}, y_{<i}, w) p(w_i^{\text{pos}} \mid w_i^{\text{arc}}, y_{<i}, w) p(w_i^{\text{morph}} \mid w_i^{\text{arc}}, y_{<i}, w).} && (2.2) \end{aligned}$$

3 Encoder-Decoder Setup

Let $N = \{1, \ldots, n\}$ be an index set. Given an input sentence $w = (w_i)_{i=1}^N$ we generate a sequence of *embeddings* $\boldsymbol{\omega} = (\boldsymbol{\omega}_i)_{i=1}^N$ where

$$\boldsymbol{\omega}_i = \mathbf{WordEmbed}(w_i) \oplus \mathbf{CharEmbed}(w_i) \oplus \mathbf{BertEmbed}(w_i).$$

Character-level embeddings are implemented via a CNN á la Chiu and Nichols (2016). BERT embeddings are finetuned from a BERT model pre-trained on German text by Chan et al. (2020).

Encoder: feed embeddings through a multi-layer bi-directional LSTM with skip-connections and dropout:

$$\mathbf{e} = (\mathbf{e}_i)_{i=0,\ldots,n} = \mathbf{BiLSTM}(\boldsymbol{\omega})$$

Decoder: feed embeddings through a single-layer uni-directional LSTM with dropout:

$$\mathbf{d} = (\mathbf{d}_i)_{i=1,\ldots,n} = \mathbf{LSTM}(\boldsymbol{\omega})$$

4 Bi-affine Attention Mechanism

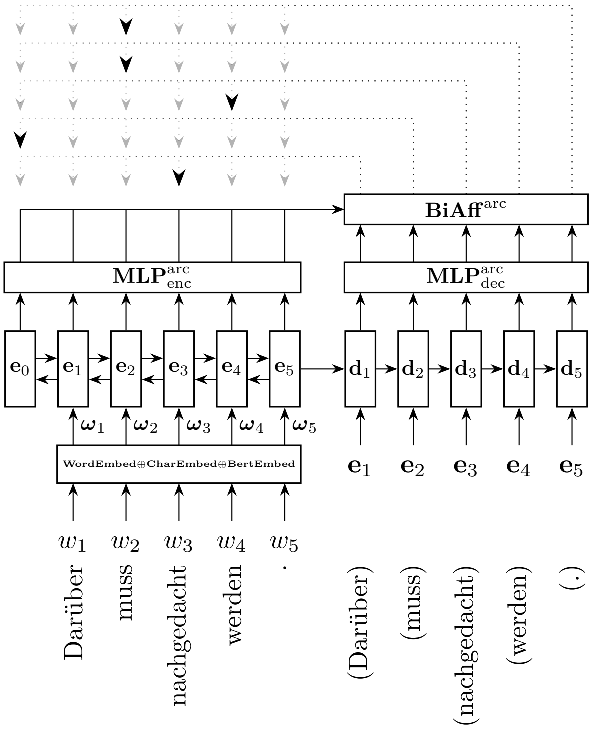
We feed $\mathbf{e}$ and $\mathbf{d}$ through MLPs to produce sequences $(\mathbf{e}^{\text{arc}}, \mathbf{d}^{\text{arc}})$ of dimension-reduced vectors. These are fed into a bi-affine layer which produces latent features $\mathbf{v}^{\text{arc}}$ that are then fed into an attention layer, resulting in logits corresponding to strength of an arc.

$$\begin{aligned} \mathbf{e}^{\text{arc}} &= \mathbf{MLP}_{\text{enc}}^{\text{arc}}(\mathbf{e}); \qquad \mathbf{d}^{\text{arc}} = \mathbf{MLP}_{\text{dec}}^{\text{arc}}(\mathbf{d}) && (4.1) \\ \mathbf{v}_{i,j}^{\text{arc}} &= \mathbf{BiAff}^{\text{arc}}(\mathbf{e}_i^{\text{arc}}, \mathbf{d}_j^{\text{arc}}) && (4.2) \\ &:= \mathbf{e}_i^{\text{arc}\mathsf{T}} \mathbf{U}_{\text{h-d}}^{\text{arc}} \mathbf{d}_i^{\text{arc}} + \mathbf{e}_i^{\text{arc}\mathsf{T}} \mathbf{U}_{\text{h-h}}^{\text{arc}} \mathbf{e}_i^{\text{arc}} + \mathbf{d}_i^{\text{arc}\mathsf{T}} \mathbf{U}_{\text{d-d}}^{\text{arc}} \mathbf{d}_i^{\text{arc}} && (4.3) \\ &\quad + U_{\text{h}}^{\text{arc}} \mathbf{e}_i^{\text{arc}} + U_{\text{d}}^{\text{arc}} \mathbf{d}_i^{\text{arc}} + \mathbf{u}_{\text{bias}}^{\text{arc}} && (4.4) \\ s_{i,j}^{\text{arc}} &= \mathbf{u}_{\text{agg}}^{\text{arc}\mathsf{T}} \tanh(\mathbf{v}_{i,j}^{\text{arc}}) && (4.5) \end{aligned}$$

Fixing dependency j, the vector $\mathbf{softmax}(\mathbf{s}_{:,j})$ can be interpreted as an estimated probability distribution over potential heads.

$$\hat{p}^{\text{arc}}(w_i \mid y_{<j}, w) = \mathbf{softmax}(\mathbf{s}_{:,j}^{\text{arc}})_i.$$

[3] If w_i^{morph} is a vector, assume every component is mutually independent with each other and with $w_i^{\text{lab}}, w_i^{\text{ord}}, w_i^{\text{pos}}$ conditional on $w_i^{\text{arc}}, y_{<i}, w$



5 Bi-affine Classifier for Attachment Order, POS and Morphology

Attachment order, POS and morphologies are predicted via a classification layer. We use a bi-affine classifier which allows us to model probabilities of classes *conditional* on arcs, and thus use structural cues in addition to encoder/decoder states to better capture the complexity of the language. The encoder and decoder are *shared* across the tasks.

For example suppose we would like to predict the part of speech $c \in \mathcal{C}$ for word w_j conditional on its parent being w_i.

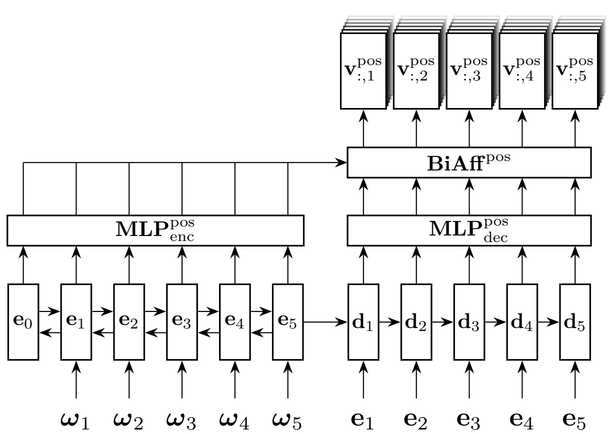
$$\mathbf{e}^{\text{pos}} = \mathbf{MLP}^{\text{pos}}_{\text{enc}}(\mathbf{e}); \qquad \mathbf{d}^{\text{pos}} = \mathbf{MLP}^{\text{pos}}_{\text{dec}}(\mathbf{d}) \tag{5.1}$$

$$\mathbf{v}^{\text{pos}}_{i,j} = \mathbf{BiAff}^{\text{pos}}(\mathbf{e}^{\text{pos}}_i, \mathbf{d}^{\text{pos}}_j) \tag{5.2}$$

$$:= \mathbf{e}^{\text{pos}\mathsf{T}}_i \mathbf{U}^{\text{pos}}_{\text{h-d}} \mathbf{d}^{\text{pos}}_i + \mathbf{e}^{\text{pos}\mathsf{T}}_i \mathbf{U}^{\text{pos}}_{\text{h-h}} \mathbf{e}^{\text{pos}}_i + \mathbf{d}^{\text{pos}\mathsf{T}}_i \mathbf{U}^{\text{pos}}_{\text{d-d}} \mathbf{d}^{\text{pos}}_i \tag{5.3}$$

$$+ U^{\text{pos}}_{\text{h}} \mathbf{e}^{\text{pos}}_i + U^{\text{pos}}_{\text{d}} \mathbf{d}^{\text{pos}}_i + \mathbf{u}^{\text{pos}}_{\text{bias}} \tag{5.4}$$

$$\hat{p}^{\text{pos}}(c \mid w_i; y_{<j}, w) = \mathbf{softmax}(\mathbf{v}^{\text{pos}}_{i,j})_c \tag{5.5}$$



Bibliography

Chan, B., Möller, T., Pietsch, M., & Soni, T. (2020). *German BERT (`bert-base-german-cased`)*. Retrieved September 1, 2023, from https://huggingface.co/bert-base-german-cased

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, *4*, 357–370.

Fernández-González, D., & Martins, A. F. (2015). Parsing as reduction. *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)*, 1523–1533.