Maximum Spanning Trees Are Invariant to Temperature Scaling in Graph-based Dependency Parsing

Stefan Grünewald
Institut für Maschinelle Sprachverarbeitung, University of Stuttgart
Bosch Center for Artificial Intelligence, Renningen, Germany
stefan.gruenewald@de.bosch.com

Abstract

Modern graph-based syntactic dependency parsers operate by predicting, for each token within a sentence, a probability distribution over its possible syntactic heads (i.e., all other tokens) and then extracting a maximum spanning tree from the resulting log-probabilities. Nowadays, virtually all such parsers utilize deep neural networks and may thus be susceptible to miscalibration (in particular, overconfident predictions). In this paper, we prove that *temperature scaling*, a popular technique for post-hoc calibration of neural networks, cannot change the output of the aforementioned procedure. We conclude that other techniques are needed to tackle miscalibration in graph-based dependency parsers in a way that improves parsing accuracy.

1 Introduction

Syntactic dependency parsing refers to the task of predicting, for a given sentence, the grammatical relations between its tokens. Most commonly, the output is a *dependency tree*, i.e., a graph structure in which each token constitutes a node and is assigned exactly one parent (its syntactic head). The parent may be either one of the other words in the sentence or an additional, implicit ROOT node. For the dependency tree to be valid, each token must be reachable from ROOT.[1] Figure 1 shows such a structure for the sentence "Mary likes fluffy cats."[2]

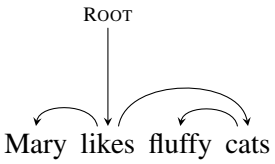

Figure 1: Example dependency tree.

	ROOT	Mary	likes	fluffy	cats
Mary	0.01	0.02	0.88	0.07	0.02
likes	0.95	0.01	0.00	0.03	0.01
fluffy	0.09	0.13	0.05	0.02	0.71
cats	0.03	0.10	0.74	0.12	0.01

Figure 2: Syntactic head probabilities.

Graph-based dependency parsing is a technique for predicting dependency trees. In its usual formulation, the approach entails using some machine learning classifier to predict, for each token within the input sentence, a probability distribution over its possible syntactic heads (i.e., all other tokens in the sentence, as well as ROOT), as shown in the rows of Figure 2. In a second step, the logarithms of the resulting probabilities are then interpreted as edge weights between pairs of nodes corresponding to tokens and a maximum spanning tree is extracted, using, e.g., the Chu-Liu/Edmonds algorithm (Chu and Liu, 1965; Edmonds, 1967). This maximum spanning tree is then returned as the dependency tree for the input sentence.

The *calibration* of a machine learning classifier refers to its ability to generate output probabilities that are representative of the actual correctness likelihoods. For example, in a well-calibrated dependency

[1] In practice, there is often the additional constraint that ROOT can only have one outgoing edge. We will discuss this later in the paper.

[2] Edges in a dependency tree may additionally be labelled according to the grammatical functions they represent. For example, because *cats* is the grammatical object of *likes*, the edge between them may be labelled *obj*. We will ignore edge labels for the remainder of this paper, as they are not relevant to our discussion.

parser, roughly 80 % of edges that are predicted with a probability of 0.8 should actually be present in the gold data. However, as Guo et al. (2017) show, modern neural networks are often miscalibrated and prone to overconfident predictions. In the context of graph-based dependency parsing, this would mean that the probability distributions over syntactic heads are too concentrated on the parent considered the most likely, not properly reflecting the uncertainty in the syntactic attachment of the tokens. Since the final dependency trees are extracted using an MST algorithm that operates on the (log-)probabilities returned by the classifier (and not merely the highest-scoring parent for each token), overconfidence in the syntactic attachments of the tokens might not only lead to overconfidence in the parse itself, but also to incorrect parses. Thus, good estimates of token attachment uncertainty may not only improve the uncertainty estimates of the final trees, but also their accuracy.

One popular technique for post-hoc calibration of neural networks is *temperature scaling* (Guo et al., 2017). This approach works by dividing the unnormalized output values ("logits") of the neural network by a constant $T \in \mathbb{R}_{>0}$ (the "softmax temperature") before applying the softmax function, i.e., given a vector $\mathbf{x}$ of unnormalized scores, the vector of probabilities is computed as $\text{softmax}(\mathbf{x}/T)$ rather than simply $\text{softmax}(\mathbf{x})$. T is optimized w.r.t. negative log-likelihood on a validation set and usually $T > 1$, resulting in a "smoother" probability distribution (i.e., one with higher entropy, or higher uncertainty) that corrects for the model's overconfidence.

In this paper, we show that applying temperature scaling to the logits of a graph-based dependency parser (i.e., the unnormalized values of the head probability distributions; the input values for the rows in Figure 2) does not change the resulting dependency trees. Although it is already known that temperature scaling does not change predictions for individual classifications, this is nonetheless a somewhat surprising result since the maximum spanning trees in dependency parsing depend on *all* (log-)probabilities provided by the underlying classifier. We conclude that calibration methods that simply re-scale softmax probabilities are not suited to improve the accuracy of graph-based dependency parsers.

The remainder of this paper is structured as follows. Section 2 sets up definitions for temperature scaling in graph-based dependency parsing; Section 3 proves our main claim; Section 4 concludes the paper.

2 Definitions

Let $V = \{v_1, \dots v_n\}$ be a set of nodes. In practice, these are the individual tokens of the sentence, as well as ROOT.

Let $X \in \mathbb{R}^{n \times n}$ a matrix of scores between nodes.[3] These are the unnormalized output values ("logits") of the dependency classifier.

Given X and a "softmax temperature" $T \in \mathbb{R}_{>0}$, we define a parametrized weight function $w_{X,T} : V^2 \mapsto \mathbb{R}$ as

$$w_{X,T}(v_i, v_j) = \log\left(\frac{e^{\frac{x_{ij}}{T}}}{\sum_{k=1}^{n} e^{\frac{x_{kj}}{T}}}\right) \tag{1}$$

$$= \log\left(e^{\frac{x_{ij}}{T}}\right) - \log\left(\sum_{k=1}^{n} e^{\frac{x_{kj}}{T}}\right) \tag{2}$$

$$= \frac{x_{ij}}{T} - \log\left(\sum_{k=1}^{n} e^{\frac{x_{kj}}{T}}\right) \tag{3}$$

$w_{X,T}$ assigns weights to all edges between two nodes; for each node v_j, the weights of incoming edges are the log-softmax values of the unnormalized scores in the corresponding row of X. For $T = 1$, this is the standard approach in graph-based dependency parsing.

[3] In practice, the matrix row corresponding to the ROOT token is often ommitted, i.e., $X \in \mathbb{R}^{(n-1)\times n}$ (see Figure 2). This is because ROOT will always constitute the root of the dependency tree and thus not have a parent of its own, making the weights of incoming edges irrelevant. However, this has no impact on our overall argument.

$D = \langle V, V^2 \rangle$ and $w_{X,T}$ form a complete, directed, weighted graph. Given D and a designated root node $r \in V$, an *arborescence* (directed spanning tree) is a subgraph $A = (r, V, E)$ with $E \subseteq V^2$ such that (a) each non-root node has exactly one incoming edge, and (b) A has no cycles.

Without loss of generality, assume that $r = v_1$. An arborescence A then induces a predecessor function $\pi : \{2, \ldots, n\} \mapsto \{1, \ldots, n\}$ that maps the index of each non-root node to the index of its parent. We can now define the weight of A as the sum of its edges:

$$w_{X,T}(A) = \sum_{i=2}^{n} w_{X,T}(v_{\pi(i)}, v_i) \tag{4}$$

and we call A a maximum arborescence of (D, r) and $w_{X,T}$ if $w_{X,T}(A) \geq w_{X,T}(A')$ for all arborescences A' of (D, r).

3 Proof of Main Claim

Our main claim is that a maximum arborescence A of (D, r) and some $w_{X,T}$ is also a maximum arborescence of (D, r) and any other $w_{X,T'}$. To prove this, we first prove the following, more general statement:

Theorem 1. *If A and A' are arborescences of (D, r) and $w_{X,T}(A) \geq w_{X,T}(A')$ for some $T \in \mathbb{R}_{>0}$, then $w_{X,T'}(A) \geq w_{X,T'}(A')$ for all $T' \in \mathbb{R}_{>0}$.*

Proof. We note the following equality:

$$\begin{aligned}
w_{X,T}(A) - w_{X,T}(A') &= \sum_{i=2}^{n} w_{X,T}(v_{\pi(i)}, v_i) - \sum_{i=2}^{n} w_{X,T}(v_{\pi'(i)}, v_i) && (5)\\
&= \sum_{i=2}^{n} \left(w_{X,T}(v_{\pi(i)}, v_i) - w_{X,T}(v_{\pi'(i)}, v_i) \right) && (6)\\
&= \sum_{i=2}^{n} \left(\frac{x_{\pi(i),i}}{T} - \log \left(\sum_{k=1}^{n} e^{\frac{x_{ki}}{T}} \right) - \frac{x_{\pi'(i),i}}{T} + \log \left(\sum_{k=1}^{n} e^{\frac{x_{ki}}{T}} \right) \right) && (7)\\
&= \sum_{i=2}^{n} \left(\frac{x_{\pi(i),i}}{T} - \frac{x_{\pi'(i),i}}{T} \right) && (8)\\
&= \frac{1}{T} \sum_{i=2}^{n} \left(x_{\pi(i),i} - x_{\pi'(i),i} \right) && (9)
\end{aligned}$$

Therefore, if $w_{X,T}(A) - w_{X,T}(A') \geq 0$, then $w_{X,T'}(A) - w_{X,T'}(A') \geq 0$ for all $T' \in \mathbb{R}_{>0}$. Equivalently, if $w_{X,T}(A) \geq w_{X,T}(A')$, then $w_{X,T'}(A) \geq w_{X,T'}(A')$ for all $T' \in \mathbb{R}_{>0}$. □

We can now use the above result to prove our main claim:

Theorem 2. *A maximum arborescence A of (D, r) for a given $w_{X,T}$ is also a maximum arborescence of (D, r) for any other $w_{X,T'}$ (with $T, T' \in \mathbb{R}_{>0}$).*

Proof. Since A is a maximum arborescence of (D, r) and $w_{X,T}$, it holds by definition that $w_{X,T}(A) \geq w_{X,T}(A')$ for all arborescences A' of (D, r). From Theorem 1 it follows immediately that $w_{X,T'}(A) \geq w_{X,T'}(A')$ for all A' also for any other $w_{X,T'}$. □

We have thus proven that applying temperature scaling to the logits of a graph-based dependency parser does not change the dependency tree returned by the overall system (assuming that maximum overall edge weight is used as the selection criterion).

Additional structural constraints. In practice, additional structural constraints are often imposed on dependency trees; most commonly, there may be only one edge emanating from the root node of the tree (Zmigrod et al., 2020). We note that our result is unaffected by these situations, as in this case, A is simply a maximum arborescence chosen from a restricted set (i.e., all arboresences that also fulfill the additional structural criterion). However, Theorem 1 still applies for all members of this restricted set, meaning that A will still be the maximum arborescence from this set for any $w_{X,T'}$.

4 Conclusion

In this paper, we have proven that temperature scaling, a popular technique for post-hoc calibration of neural network classifiers, does not have any effect on the output of graph-based syntactic dependency parsers. We conclude that more sophisticated methods – in particular, those that may also change class predictions instead of only re-scaling output probabilities – may be needed for tackling miscalibration in dependency parsers in ways that may lead to improved accuracy. Investigating such methods could be a promising direction for future work.

Acknowledgements

The author thanks Annemarie Friedrich and Sophie Henning for their helpful suggestions and feedback.

References

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330.

Ran Zmigrod, Tim Vieira, and Ryan Cotterell. 2020. Please mind the root: Decoding arborescences for dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4809–4819, Online, November. Association for Computational Linguistics.