Proof of Theorem 1

Tao Lei
CSAIL,MIT

Here we give the proofs of Theorem 1 and other necessary lemmas or corollaries.

Lemma 1 (Reachability) *Any two trees y, y' are reachable to each other. Specifically, let $m_1, m_2, \cdots, m_n$ be the bottom-up list of nodes in tree y, then there exists a path $y = y^{(0)} \to y^{(1)} \to \cdots \to y^{(n)} = y'$, in which $y^{(i)}$ is obtained by changing the head of m_i, i.e. $y^{(i)}(m_i) = y'(m_i)$, and this change* ***always*** *results in a valid tree (which has no circle).*

Proof: We show $y^{(i)}$ is always a valid tree and therefore $y^{(i)} \in \mathcal{T}(y^{(i-1)}, m_i)$, because $y^{(i)}$ and $y^{(i-1)}$ differs at most at the head of m_i ($y^{(i-1)}(m_i) = y(m_i)$ but $y^{(i)}(m_i) = y'(m_i)$). Proof by induction on $i = 1, \cdots, n$.

- **i=1:** m_1 must be leaf node in y because $m_1, \cdots, m_n$ is a bottom-up (reverse DFS) order. Changing its head to any node cannot results in a circle. Therefore $y^{(i)}$ is a tree when $i = 1$.

- **i>1:** Now let's change the head of m_i in tree $y^{(i-1)}$. Consider the subtree with root m_i in $y^{(i-1)}$. We now prove that any node x inside this subtree is already processed and its head is already changed, i.e., $x \in \{m_1, \cdots, m_{i-1}\}$ and $y^{(i-1)}(x) = y'(x)$. This can be shown by contradiction. Assume a node x inside this subtree is not processed, and its head h has not been changed yet, i.e., $y(x) = y^{(i-1)}(x) = h$. This implies the node h has not been processed neither, because all nodes are processed in bottom-up order. Repeat the same idea and we know that $x, y(x), y(y(x)), \cdots, y(y(..y(x)..)) = m_i$ are not processed. This contradicts to the fact that m_i is the next immediate unprocessed node in the bottom-up list, because its descendants are not processed.

Function **CountOptima**($G = \langle V, E \rangle$)

$V = \{w_0, w_1, \cdots, w_n\}$ are the root (w_0) and n words

$E = \{e_{ij} \in \mathbb{R}\}$ are the arc scores

Return: the number of local optima

Let $y(0) = \emptyset$ and $y(i) = \arg\max_j e_{ji}$;

if y is a tree (no circle) **then return** 1;

Find a circle $C \subset V$ in y;

cnt $= 0$;

// contract the circle

create a vertex w_*;

$\forall j \notin C : e_{*j} = \max_{k \in C} e_{kj}$;

for each vertex $w_i \in C$ **do**

  $\forall j \notin C : e_{j*} = e_{ji}$;

  $V' = V \cup \{w_*\} \setminus C$;

  $E' = E \cup \{e_{*j}, e_{j*} \mid \forall j \notin C\}$;

  cnt += CountOptima($G' = \langle V', E' \rangle$);

end for

return cnt;

Figure 1: A recursive algorithm for counting local optima for a sentence with words $w_1, \cdots, w_n$ in first-order case. The idea is very similar to the Chu-Liu-Edmond algorithm for finding only the maximum directed spanning tree.

So all nodes in the subtree are processed and all arcs appear in the subtree are already arcs in y'. Changing the head of m_i cannot results in a circle (i.e., the new head of m_i, $y'(m_i)$ can not be a node inside this subtree, otherwise it implies there is a circle in tree y', which is not possible). Thus $y^{(i)}$ is a valid tree.

Finally $y^{(n)} = y'$ because $y^{(n)}(x) = y'(x)$ for all node x. In sum, y' is accessible via the path $y = y^{(0)} \rightarrow \cdots \rightarrow y^{(n)} = y'$. ∎

To prove Theorem 1, we start by proving the correctness of the recursive algorithm for counting local optima:

Definitions Let $G = \langle V, E \rangle$ be a directed weighted graph of size $n+1$, where vertices $V = \{w_0, \cdots, w_n\}$ represent a pseudo root node w_o and n words $w_1, \cdots, w_n$ in a sentence, and weights $E = \{e_{ij} \in \mathbb{R}\}$ represent the first-order scores associated with individual arcs $i \to j$. A local optimum tree in G is a directed tree with root w_0, such that changing any **single** head cannot result in a better tree with higher score.[1]

Lemma 2 *Let $y(0) = \emptyset$ and $y(i)$ be the index of the best possible head for word w_i, i.e., $y(i) = \arg\max_j e_{ji}$. Then:* ***(a)*** *y is the unique local optimum in G if y is a tree;* ***(b)*** *otherwise let C be a circle in y, then any local optimum tree $\tilde{y} \in G$ contains exactly $|C| - 1$ arcs in the circle C.*

Proof: **(a)** Simply by the definition of y.

(b) Proof by contradiction. Assume $\tilde{y} \in G$ is a local optimum tree that contains less than $|C| - 1$ arcs in the circle C. Consider a top-down order of nodes in $\tilde{y}$, and let $u \in C$ be the **first** node (in the circle) in this top-down list. Now define $\hat{y}$ as follows,

$$\hat{y}(x) = \begin{cases} \tilde{y}(x) & x \notin C \text{ or } x = u \\ y(x) & x \in C \text{ and } x \neq u \end{cases}$$

It's easy to verify that $\hat{y}$ is a tree. Note that $\hat{y}$ has exactly $|C| - 1$ arcs of the circle C. By Lemma 1, there is a path from $\tilde{y}$ to $\hat{y}$ that never decreases the tree score, because the heads of $\hat{y}$ is strictly better than those of $\tilde{y}$, i.e., $e_{\hat{y}(x)x} \geq e_{\tilde{y}(x)x}$. This contradicts to the assumption that $\tilde{y}$ is a local optimum tree. ∎

Now according to Lemma 2, one way to get the local optimum trees in G is as follows: (1) enumerate and pick a node $u \in C$; (2) remove the arc $y(u) \to u$ in the circle C and it becomes a chain; (3) fix these heads and arcs in the chain; (4) contract this chain and search for local optima in a smaller graph by applying Lemma 1 repeatedly:

[1] We assume there is no tie when comparing scores, trees or heads. If there is a tie, we can always break it by taking the tree (or head) that ranks higher in terms of aphabetic order.

Definitions Let G, y and C be a graph, the set of best heads and the circle in y respectively. Without loss of generality, let $w_1, \cdots, w_c$ be the nodes in the circle C, where $c = |C|$. Define graph $G^{(i)} = \left\langle V^{(i)}, E^{(i)} \right\rangle (i = 1, \cdots, c)$ as the **contraction** of graph G at $w_i \in C$ as follows:

$$\begin{aligned} V^{(i)} &= V \cup \{w_*^{(i)}\} \setminus C \\ E^{(i)} &= \{e'_{jk}\} \end{aligned}$$

where

$$\begin{aligned} e'_{j*} &= e_{ji}, & \forall j \in V \setminus C \\ e'_{*j} &= \max_{k \in C} e_{kj} & \forall j \in V \setminus C \\ e'_{jk} &= e_{jk} & \forall j, k \in V \setminus C \end{aligned}$$

Lemma 3 *Any local optimum tree $\tilde{y} \in G^{(i)}$ is also a local optium tree in G (by uncontracting the node w_* back to the chain); and vice versa, i.e., any local optimum tree $\tilde{y} \in G$ is also a local optimum tree in one of $G^{(i)}$ for $i = 1, \cdots, c$.*

Proof: By Lemma 2 and the definitions of $G^{(i)}$ and y. Details omitted here. ▮

Corollary 1 *Let $F(G)$ be the number of local optimum tree in graph G: (a) $F(G) = 1$ if y is a tree that has no circle; (b) $F(G) = \sum_i F(G^{(i)})$ if y contains a circle C.*

Proof: By Lemma 2 and Lemma 3. ▮

Corollary 2 *The recursive algorithm in Figure 1 returns the number of local optima in graph G. Its complexity is linear to the number of local optima.*

Proof: By Lemma 2, Lemma 3 and Corollary 1. ▮

Theorem 1 (Local Optima Bound) *For any first-order score function that factorizes into the sum of arc scores $S(x, y) = \sum S_{arc}(y(m), m)$: (a) the number of local optimum trees is at most 2^{n-1} for n words; (b) this upper bound is tight.*

Proof: **(a)** Let $\hat{F}(m)$ be the maximum number of local optimum trees in **any** graph of size m. By Corollary 1, we have:

$$\begin{aligned} &\hat{F}(2) = 1 \\ &\hat{F}(m) \leq \max_{2 \leq c \leq m-1} \hat{F}(m - c + 1) \times c \qquad\qquad \forall m > 2 \end{aligned}$$

Solving this we get $\hat{F}(m) \leq 2^{m-2}$. For a sentence with n words, the corresponding graph has size $m = n + 1$, therefore the upper bound is 2^{n-1}.

(b) For any $n > 0$, construct a graph $G_n = \langle V, E \rangle$ as follows:

$$\begin{aligned} V &= \{w_0, w_1, \cdots, w_n\} \\ E &= \{e_{ij}\} \end{aligned}$$

where

$$e_{ij} = e_{ji} = i \qquad\qquad \forall 0 \leq i < j \leq n$$

Note that $w_{n-1} \rightarrow w_n \rightarrow w_{n-1}$ is a circle of length 2 in G_n and y. Then it can be shown by induction on n and Corollary 1 that $F(G_n) = F(G_{n-1}) \times 2 = 2^{n-1}$. ∎