写真列と構造要素からの手順構造と手順書の同時学習

西村 太一[1]，橋本 敦史[2]，牛久 祥孝[2]，森 信介[3]
[1] 京都大学大学院 情報学研究科，[2] オムロンサイニックエックス株式会社，
[3] 京都大学 学術情報メディアセンター

nishimura.taichi.43x@st.kyoto-u.ac.jp,
{atsushi.hashimoto, yoshitaka.ushiku}@sinicx.com, forest@i.kyoto-u.ac.jp

1 はじめに

Smart Kitchen [4] や料理ロボット [1] のような，作業状態を認識し，人と協調して作業を進めるシステムにおいて，コンピュータが作業内容を言語で伝達することは作業中でも有益な伝達手段である．作業内容を生成するためには，コンピュータは初期状態から現在の状態に至るまでの手続き的な構造を理解することが求められる．この構造をグラフや木構造で表現する研究 [8, 11, 14] は行われてきたものの，それを加味して手順書を生成する取り組みは未だ行われていない．我々は，この構造を取り込むことが実用的な精度で手順書を得るために必要であると考える．そして，構造予測と手順書生成の両タスクには依存関係があり，構造を予測しながら手順書を生成するように構造と手順書を同時学習する枠組みを利用することで，両タスクにおいて精度を向上できるのではないかと仮定した．

以上の仮定に基づき，本研究では料理ドメインを対象に作業の初期状態として材料列を，中間状態として写真列を与え，その構造とレシピ (手順書) を同時生成するという課題を提案する (図 1)．本論文では，この構造を木構造として表現する既存研究 [14] を参考に，visual SIMMR (vSIMMR) という新しいデータセットを作成する．このデータセットは，写真列，写真列中で材料がどのように使われるのかを木構造で表現した材料木，写真列の対となる手順列の 3 つ組の集合のデータセットである．このデータセットを用いて，写真列と材料列をもとに材料木の構造を予測し，予測結果をもとに手順書を生成する手法を提案する．加えて，生成される手順書と，それを実行した結果である写真列は同じ構造を持つという仮定のもと，得られた手順書から再度材料木を予測する．実験の結果，同時学習を行うことで材料木の構造予測，手順書の生成において同時学習を行わない場合と比較して両タスクにおいて精度が向上することを示した．また，生成した手順書から再度材料木を予測することで，さらに精度が向上することを確認した．

2 visual SIMMR (vSIMMR)

vSIMMR は，写真列，写真列中にどのように材料が使われるのかを木構造で表現した材料木，写真列の対となる手順列の 3 つ組の集合のデータセットである．図 1 の材料木がその一例である．この例では，手順 1 の中間ノードはトマトを入力にして手順を実施して得

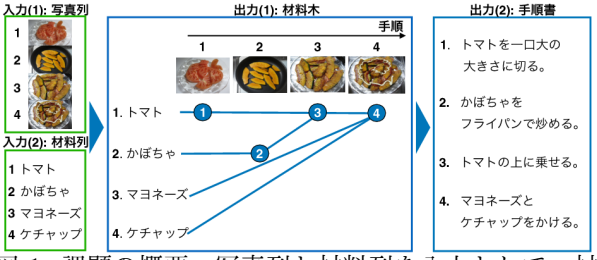

図 1: 課題の概要．写真列と材料列を入力として，材料木と手順書を生成する．

られた出力を表していることや，手順 3 の中間ノードは切られたトマトと，手順 2 の炒められたかぼちゃから得られるものであるということが分かる．このように，材料木では各手順は 1 つの中間ノード持ち，最後の手順が完成物を表す．この時，全ての材料が使われていなければならない．　vSIMMR のアノテーション対象としたのは Cookpad Image Dataset [3] のレシピである．この中から，全ての手順に写真がついており，かつ材料数と手順数が 3 以上のものを対象に，以下のプロセスでアノテーションを行なった．まず，材料を正規化する．アノテーションするレシピの材料はユーザが書いているものであり，誤字脱字 (例: “アボガド”(誤)，“アボカド”(正)) や表記揺れ (例: “玉ねぎ”，“玉葱”，“たまねぎ”) も含まれるため，これらを手動で訂正した．次に，書き直した材料を写真列の各写真で使われる順番に並べ替えた．この時，作業に関係しない手順 (例: “話題のレシピになりました。”) や，材料に関係しない手順 (例: “鍋を熱しておきます。”) は手順書から除外した．最後に並べ替えた材料と写真列，レシピをもとに写真列の各写真を参照しながら材料木をアノテーションした．この時，調味料のように料理を完成させる上で必要不可欠であるが，画像から判断するのは難しいものについては，画像からだけでなく手順書を参照して判断すべきと指示した．また，vSIMMR で表現できないレシピは今回アノテーションを行なっていない．例えば，木構造では材料が作業の途中で分離することは表現できない (例: “玉子の黄身と白身を分ける。”)．

表 1 に構築した vSIMMR データセットの統計情報を示す．手順書においては，訓練データで現れる頻度が 3 回以下の単語は未知語として処理し，語彙サイズを測定した．

表 1: vSIMMR の統計情報.

	訓練	検証	評価
レシピ数	1,603	250	250
平均手順数	6.78	6.74	6.85
平均単語数	118.23	113.91	114.68
平均材料数	6.58	6.37	6.64
語彙サイズ		2,842	

3 提案手法

図 2 に提案手法の概要を示す．提案手法は 4 つのプロセスからなる．(i) 最初に，写真列と材料列を入力として材料木を予測する．(ii) 次に，得られた予測結果からサンプリングを行い，材料木を生成し，Tree-LSTM [13] を用いて材料木の中間ノードを特徴ベクトルに変換する．(iii) 得られた材料木の各中間ノードと写真列の各写真をそれぞれ結合し，Encoder-Decoder モデルを用いて手順書を生成する．(iv) 最後に，生成した手順書と写真列から得られる材料木は同じ構造となるという仮定に基づき，手順書から材料木を再度予測する．全てのモジュールは微分可能な形で演算を行うモジュールのみで構築されているため，End-to-end に同時学習することができる．以下では，各プロセスを一つ一つ説明する．

(i) 材料木予測: 材料木は，写真列と材料列を入力とし，ある材料がどの手順で使われるのかを確率で表す材料-手順隣接行列 $\boldsymbol{X}$ と，ある手順の結果得られたものがどの手順で使われるのかを同じく確率で表す手順-手順隣接行列 $\boldsymbol{Y}$ を直接予測する形で生成する．本研究では，この 2 つの隣接行列は互いに関係があると考え，先に手順-手順隣接行列を生成する時に得られる特徴ベクトルを用いて材料-手順隣接行列に活用することとした．そのためにまず，手順-手順隣接行列を計算する．写真列の各写真を，訓練済みの画像エンコーダ ψ を用いてエンコードすることで，各写真のベクトル表現 $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k, \ldots, \boldsymbol{v}_K)$ を得る．そして，$\boldsymbol{V}$ を 2 つの異なる biLSTM である，biLSTM$_1$ と，biLSTM$_2$ に入力し，$\dot{\boldsymbol{V}} = (\dot{\boldsymbol{v}}_1, \dot{\boldsymbol{v}}_2, \ldots, \dot{\boldsymbol{v}}_k, \ldots, \dot{\boldsymbol{v}}_K)$ と，$\ddot{\boldsymbol{V}} = (\ddot{\boldsymbol{v}}_1, \ddot{\boldsymbol{v}}_2, \ldots, \ddot{\boldsymbol{v}}_k, \ldots, \ddot{\boldsymbol{v}}_K)$ を得る．こうして得られた 2 つの biLSTM の出力結果を用いて，手順-手順隣接行列を以下のように得る．

$$\boldsymbol{Y}_{i,j} = \frac{\exp(\dot{\boldsymbol{v}}_i^T \ddot{\boldsymbol{v}}_j)}{\sum_{\ell=1}^{\ell=K} \exp(\dot{\boldsymbol{v}}_i^T \ddot{\boldsymbol{v}}_\ell)} \tag{1}$$

次に，材料-手順隣接行列を計算する．前述したように，手順-手順隣接行列を生成する際に得られた 2 つの特徴ベクトルを利用するために，$\dot{\boldsymbol{V}}$ と $\ddot{\boldsymbol{V}}$ の各ベクトルの間で，各次元において大きい値を計算し，画像行列 $\hat{\boldsymbol{V}}$ を得る．材料列から，各材料を word2vec [10] を用いて分散表現に変換した後，LSTM へ入力して材料ごとの分散表現を獲得し，各材料の特徴ベクトル $\boldsymbol{G} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_n, \ldots, \boldsymbol{g}_N)$ を得る．これを，biLSTM$_3$ に入力することで得る材料行列 $\hat{\boldsymbol{G}} = (\hat{\boldsymbol{g}}_1, \hat{\boldsymbol{g}}_2, \ldots, \hat{\boldsymbol{g}}_n, \ldots, \hat{\boldsymbol{g}}_N)$ と，画像行列 $\hat{\boldsymbol{V}}$ を用いて，材料-手順隣接行列を以下のように計算する．

$$\boldsymbol{X}_{i,j} = \frac{\exp(\hat{\boldsymbol{g}}_i^T \hat{\boldsymbol{v}}_j)}{\sum_{\ell=1}^{\ell=K} \exp(\hat{\boldsymbol{g}}_i^T \hat{\boldsymbol{v}}_\ell)} \tag{2}$$

得られた予測結果と正解の隣接行列を用いて，材料木予測の損失関数 $\mathcal{L}_T$ を交差エントロピー誤差として計算する．

(ii) 材料木の生成とエンコード: 得られた隣接行列 $\boldsymbol{X}, \boldsymbol{Y}$ から，微分可能な形で材料木を得るために，Gumbel softmax [7] を用いてサンプリングを行う．このサンプリングを経て得られた木構造に対し，Tai らによって提案された Tree-LSTM [13] の一つである Child-sum LSTM を用いて木構造を保持しながら k 番目の手順を特徴ベクトル $\boldsymbol{h}_k$ へ変換する．Child-sum LSTM では，k 番目の手順行列のベクトル $\hat{\boldsymbol{v}}_k$ と，k 番目の中間ノードに接続している子ノードの集合 $C(k)$ を用いて，以下のように k 番目の隠れ層 $\boldsymbol{h}_k$ とメモリセル $\boldsymbol{c}_k$ を更新する．この時，初期状態の隠れ層として，材料行列の各材料ベクトル $\hat{\boldsymbol{g}}_n$ から対応する隠れ層 $\boldsymbol{h}_0^n$ とメモリセル $\boldsymbol{c}_0^n$ を計算し，Child-sum LSTM の初期状態として与える．

$$\hat{\boldsymbol{h}}_k = \sum_{j \in C(k)} \boldsymbol{h}_j^k \tag{3}$$

$$\boldsymbol{i}_k = \sigma(\boldsymbol{W}^{(i)} \boldsymbol{x}_k + \boldsymbol{U}^{(i)} \hat{\boldsymbol{h}}_k + \boldsymbol{b}^{(i)}) \tag{4}$$

$$\boldsymbol{f}_{kj} = \sigma(\boldsymbol{W}^{(f)} \boldsymbol{x}_k + \boldsymbol{U}^{(f)} \hat{\boldsymbol{h}}_k + \boldsymbol{b}^{(f)}) \tag{5}$$

$$\boldsymbol{o}_k = \sigma(\boldsymbol{W}^{(o)} \boldsymbol{x}_k + \boldsymbol{U}^{(o)} \hat{\boldsymbol{h}}_k + \boldsymbol{b}^{(o)}) \tag{6}$$

$$\boldsymbol{u}_k = \tanh(\boldsymbol{W}^{(u)} \boldsymbol{x}_k + \boldsymbol{U}^{(u)} \hat{\boldsymbol{h}}_k + \boldsymbol{b}^{(u)}) \tag{7}$$

$$\boldsymbol{c}_k = \boldsymbol{i}_k \odot \boldsymbol{u}_k + \sum_{j \in C(k)} \boldsymbol{f}_{kj} \odot \boldsymbol{c}_j \tag{8}$$

$$\boldsymbol{h}_k = \boldsymbol{o}_k \odot \tanh(\boldsymbol{c}_k) \tag{9}$$

(iii) 手順書生成: Tree-LSTM によって得られた k 番目の隠れ層 $\boldsymbol{h}_k$ と，写真列の各写真ベクトル $\boldsymbol{v}_k$ を結合し，Encoder-Decoder モデルに入力することで，手順書中の k 番目の手順を生成する．この時，微分可能な形で手順書を生成するために，Gumbel softmax を用いて単語を出力する．こうして得られた手順列を手順書として出力する．得られた手順書の予測結果と，正解の手順書を用いて手順書生成の損失関数 $\mathcal{L}_P$ を交差エントロピー誤差として計算する．

(iv) 材料木の再予測: 我々は，手順書から得られる材料木と写真列から得られる材料木は同じ構造であると仮定し，生成した手順書から再度材料木を生成することで，材料木の予測，手順書の生成に関して精度向上が期待できると考え，以下のように材料木の再予測を行なった．最初に，各手順から手順単位での特徴ベクトルを抽出する．k 番目の手順の単語列を biLSTM$_4$ へ入力し，最初と最後の隠れ層を結合することで k 番目の手順の特徴ベクトルを得た．次に，手順書全体の時系列を考慮するために，得られた手順単位の特徴ベクトルを biLSTM$_5$ へ入力し，手順書を全体を考慮した手順ベクトルからなる手順行列 $\boldsymbol{L} = (\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_k, \ldots, \boldsymbol{l}_K)$ を得る．この手順行列 $\boldsymbol{L}$ と，材料木を予測する際に得

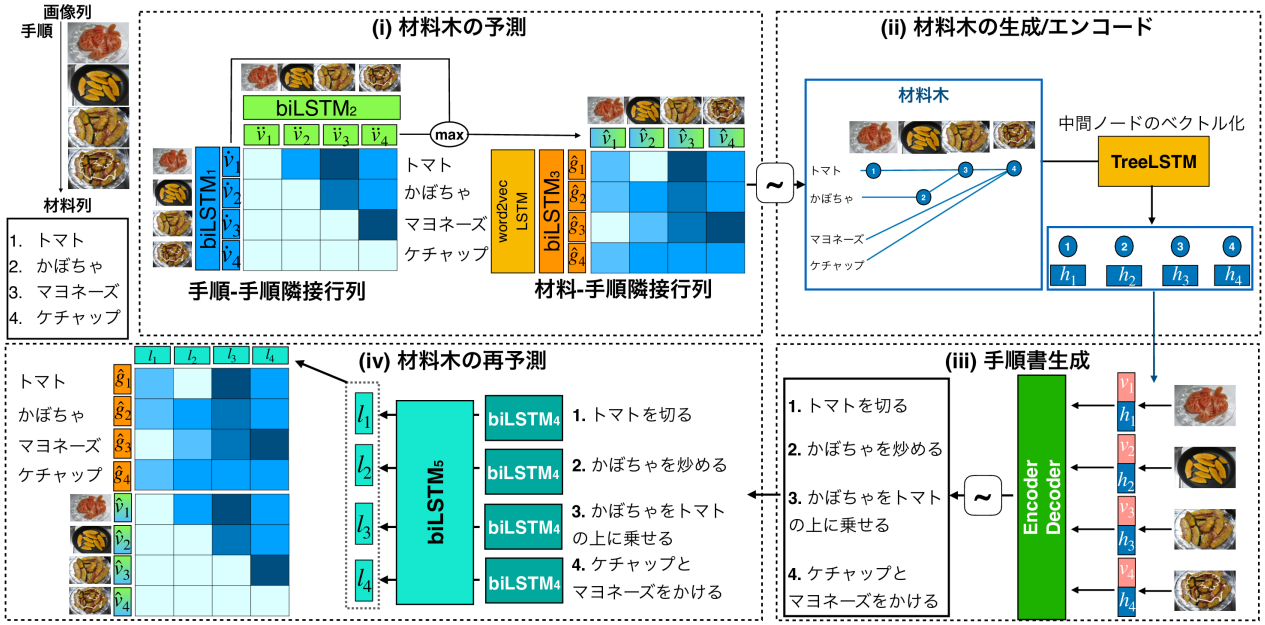

図 2: 提案手法の概要.

られた材料行列 $\hat{\boldsymbol{G}}$，画像行列 $\hat{\boldsymbol{V}}$ を用いて再び材料-手順行列 $\tilde{\boldsymbol{X}}$ と，手順-手順行列 $\tilde{\boldsymbol{Y}}$ を予測する．

$$\tilde{\boldsymbol{X}}_{i,j} = \frac{\exp(\hat{\boldsymbol{g}}_i^T \boldsymbol{l}_j)}{\sum_{\ell=1}^{\ell=K} \exp(\hat{\boldsymbol{g}}_i^T \boldsymbol{l}_\ell)} \tag{10}$$

$$\tilde{\boldsymbol{Y}}_{i,j} = \frac{\exp(\hat{\boldsymbol{v}}_i^T \boldsymbol{l}_j)}{\sum_{\ell=1}^{\ell=K} \exp(\hat{\boldsymbol{v}}_i^T \boldsymbol{l}_\ell)} \tag{11}$$

得られた予測結果と正解の隣接行列を用いて，材料木の再予測の損失関数 $\mathcal{L}_R$ を (i) と同様に交差エントロピー誤差として計算する．

全体の損失関数: モデル全体を学習するための損失関数 $\mathcal{L}$ は，得られた損失関数 $\mathcal{L}_T, \mathcal{L}_P, \mathcal{L}_R$ を用いて以下のように書く．

$$\mathcal{L} = \mathcal{L}_T + \mathcal{L}_P + \alpha \mathcal{L}_R \tag{12}$$

ただし，α はハイパーパラメータである．

4 実験

生成した材料木が文生成の性能を高めるか否かという点と，文生成を同時学習することによって材料木を予測しやすくなるか否かという二点を検証した．加えて，材料木を再予測することによって，これらの性能において更なる精度向上が見られるか否かも検証した．以下，実験設定の詳細と，実験結果を報告する．

実験設定: 画像のエンコーダ ψ には，[12] と同じエンコーダを用いた．このエンコーダは，画像を ResNet50 [5] でベクトルに変換し，得られた画像ベクトルと手順ベクトルをあらかじめ 16 万のレシピで距離学習を用いて学習したものである．こうすることで，より料理ドメインに適したベクトルが得られ，ResNet50 の画像ベクトルを利用するよりも以下の評価実験で良い性能を得ることができた．この画像のエンコーダの出力の次元数 d は 2,048 である．材料ベクトルを得るためのハイパーパラメータは，word2vec の出力の次元数を 300，材料の LSTM の次元数を 512 と設定した．そして，隣接行列を得るための全ての biLSTM の出力の次元数は全て 512 として設定した．Gumbel softmax の温度パラメータ τ は，材料木を得る時，手順書を得る時共に 0.5 として固定した．なお，損失関数中のハイパーパラメータ α は 0.01 として設定した．また，バッチサイズは 32 とし，Adam を用いて最適化を行なった．学習時，検証用データで最も BLEU4 が高かったモデルを評価するモデルとした．この実験では，手順書生成のための Encoder-Decoder として Images2seq [6]，GLACNet [9]，RetAttn [12]，SSiD [2]，SSiL [2] を利用した[1]．

自動評価尺度による評価: 材料木が文生成の性能向上に貢献するか否かを調べるために，提案手法と，Encoder-Decoder のモデルを用いて手順書をそれぞれ生成し，手順書レベルで BLEU，ROUGE-L，CIDEr-D，METEOR を評価した．この時，公平に比較するために，文生成における Encoder-Decoder のモデルには画像を入力するだけでなく材料を提案手法の材料木を予測する際に得られる材料行列へ変換し，その平均ベクトルを写真列の最初の画像ベクトルと結合させて入力した．表 2 にその結果を示す．材料木を予測しながら文生成を行うことで，どの Encoder-Decoder を用いても概ね性能が向上した．このことから，材料木を予測し利用することは手順書生成に寄与することが分かった．また，材料木を再予測することによって，さらに性能が向上し，BLEU4，ROUGE-L，CIDEr-D，METEOR で最も高い性能を得た．よって，材料木を

[1]SSiL の場合，生成文のクラスタ番号の予測結果と参照文のクラスタ番号の推移確率の間で KL ダイバージェンスを計算し，式 12 に加えている．

表 2: 自動評価尺度による評価結果.

ベースライン	BLEU1	BLEU4	ROUGE-L	CIDEr-D	METEOR
Images2seq [6]	30.2	5.1	18.0	23.9	20.8
RetAttn [12]	**37.5**	7.5	23.6	27.9	23.0
GLACNet [9]	33.2	7.4	23.8	23.6	23.2
SSiD [2]	32.9	7.1	24.4	32.3	23.3
SSiL [2]	33.8	7.3	24.7	20.2	22.3
+材料木予測					
Images2seq	32.3	5.8	18.1	25.9	21.7
RetAttn	34.9	8.6	25.1	29.5	23.6
GLACNet	35.3	8.3	23.9	25.3	23.4
SSiD	36.5	8.6	25.1	29.0	24.0
SSiL	34.9	7.6	24.2	33.0	23.3
+材料木の再予測					
Images2seq	33.0	5.3	20.8	33.1	20.3
RetAttn	37.1	**9.1**	24.6	30.4	23.6
GLACNet	37.2	8.7	**25.6**	**35.2**	**24.4**
SSiD	37.2	8.3	**25.6**	26.8	24.0
SSiL	36.4	7.9	24.9	27.9	23.8

表 3: 構造木の正解率.

	材料-手順	手順-手順
文生成なし	0.691	0.895
+手順書生成		
Images2seq	0.706	0.902
RetAttn	0.712	0.901
GLACNet	0.712	0.906
SSiD	0.701	0.904
SSiL	0.721	0.899
+材料木の再予測		
Images2seq	0.719	0.902
RetAttn	0.717	0.902
GLACNet	0.719	0.899
SSiD	0.714	**0.907**
SSiL	**0.737**	0.906

再予測することは手順書生成の性能向上に寄与していると言える.

構造木の正解率: 生成した材料木の性能を評価するために，テストセットの全ての材料木と比較して予測した材料木の正解率を計算した．表 3 にその結果を示す．手順書生成をしなかった場合に比べ，手順書を同時生成することによってより高い精度で材料木を予測することができた．このことから，材料木を生成する上で手順書を同時生成することは効果があると言える．さらに，材料木を再予測することによって，さらに高い性能で構造木を生成することができた．よって，材料木を手順書から再予測することでは，写真列から材料木を予測することにも効果があると言える.

5 おわりに

本研究では，コンピュータが作業構造を理解し発話することを目的として，料理ドメインを対象にこの構造と手順書を同時生成するという課題と，これを解決するデータセットと手法を提案した．提案するデータセットである vSIMMR は，材料を初期状態として全て葉に持ち，作業の中間状態を中間ノードが表現する木構造からなるデータセットである．提案手法では，この木構造を予測，生成しながら手順書を生成し生成した手順書から再度木構造を予測するモデルを提案した．実験の結果，木構造と手順書生成を同時学習することで，両タスクにおいて性能を向上しただけでなく，再度木構造を予測することによって，更に性能を向上したことを実験的に確認した.

参考文献

[1] Bollini et al. Interpreting and executing recipes with a cooking robot. *Experimental Robotics*, pp. 481–495, 2013.

[2] Chandu et al. Storyboarding of recipes: Grounded contextual generation. In *Proc of ACL*, pp. 6040–6046, 2019.

[3] Harashima et al. Cookpad image dataset: An image collection as infrastructure for food research. In *Proc of SIGIR*, pp. 1229–1232, 2017.

[4] Hashimoto et al. Smart kitchen: A user centric cooking support system. In *Proc of IPMU*, pp. 848–854, 2008.

[5] He et al. Deep residual learning for image recognition. In *Proc of CVPR*, pp. 770–778, 2016.

[6] Huang et al. Visual storytelling. In *Proc of NAACL*, pp. 1233–1239, 2016.

[7] Jang et al. Categorical reparameterization with gumbel-softmax. In *Proc of ICLR*, 2016.

[8] Kiddon et al. Mise en place: unsupervised interpretation of instructional recipes. In *Proc of EMNLP*, pp. 982–992, 2015.

[9] Kim et al. GLAC net: Glocal attention cascading networks for multi-image cued story generation. *CoRR*, Vol. abs/1805.10973, , 2018.

[10] Mikolov et al. Distributed representations of words and phrases and their compositionality. In *Proc of NeurIPS*, pp. 3111–3119, 2013.

[11] Mori et al. Flow graph corpus from recipe texts. In *Proc of LREC*, pp. 2370–2377, 2014.

[12] Nishimura et al. Procedural text generation from a photo sequence. In *Proc of INLG*, pp. 409–414, 2019.

[13] Tai et al. Improved semantic representations from tree-structured long short-term memory networks. In *Proc of ACL-IJCNLP*, 2015.

[14] J. Jermsurawong and N. Habash. Predicting the structure of cooking recipes. In *Proc of EMNLP*, pp. 2370–2377, 2015.