LISA: Explaining Recurrent Neural Network Judgments via Layer-wIse Semantic Accumulation and Example to Pattern Transformation

Pankaj Gupta[1,2], **Hinrich Schütze**[2]
[1]Corporate Technology, Machine-Intelligence (MIC-DE), Siemens AG Munich, Germany
[2]CIS, University of Munich (LMU) Munich, Germany
pankaj.gupta@siemens.com
pankaj.gupta@campus.lmu.de | inquiries@cislmu.org

Abstract

Recurrent neural networks (RNNs) are temporal networks and cumulative in nature that have shown promising results in various natural language processing tasks. Despite their success, it still remains a challenge to understand their hidden behavior. In this work, we analyze and interpret the cumulative nature of RNN via a proposed technique named as *Layer-wIse-Semantic-Accumulation* (LISA) for explaining decisions and detecting the most likely (i.e., saliency) patterns that the network relies on while decision making. We demonstrate (1) *LISA*: "How an RNN accumulates or builds semantics during its sequential processing for a given text example and expected response" (2) *Example2pattern*: "How the saliency patterns look like for each category in the data according to the network in decision making". We analyse the sensitiveness of RNNs about different inputs to check the increase or decrease in prediction scores and further extract the saliency patterns learned by the network. We employ two relation classification datasets: SemEval 10 Task 8 and TAC KBP Slot Filling to explain RNN predictions via the *LISA* and *example2pattern*.

1 Introduction

The interpretability of systems based on deep neural network is required to be able to explain the reasoning behind the network prediction(s), that offers to (1) verify that the network works as expected and identify the cause of incorrect decision(s) (2) understand the network in order to improve data or model with or without human intervention. There is a long line of research in techniques of interpretability of Deep Neural networks (DNNs) via different aspects, such as explaining network decisions, data generation, etc. Erhan et al. (2009); Hinton (2012); Simonyan et al. (2013) and Nguyen et al. (2016) focused on model aspects to interpret neural networks via activation maximization approach by finding inputs that maximize activations of given neurons. Goodfellow et al. (2014) interprets by generating adversarial examples. However, Baehrens et al. (2010) and Bach et al. (2015); Montavon et al. (2017) explain neural network predictions by sensitivity analysis to different input features and decomposition of decision functions, respectively.

Recurrent neural networks (RNNs) (Elman, 1990) are temporal networks and cumulative in nature to effectively model sequential data such as text or speech. RNNs and their variants such as LSTM (Hochreiter and Schmidhuber, 1997) have shown success in several natural language processing (NLP) tasks, such as entity extraction (Lample et al., 2016; Ma and Hovy, 2016), relation extraction (Vu et al., 2016a; Miwa and Bansal, 2016; Gupta et al., 2016, 2018c), language modeling (Mikolov et al., 2010; Peters et al., 2018), slot filling (Mesnil et al., 2015; Vu et al., 2016b), machine translation (Bahdanau et al., 2014), sentiment analysis (Wang et al., 2016; Tang et al., 2015), semantic textual similarity (Mueller and Thyagarajan, 2016; Gupta et al., 2018a) and dynamic topic modeling (Gupta et al., 2018d).

Past works (Zeiler and Fergus, 2014; Dosovitskiy and Brox, 2016) have mostly analyzed deep neural network, especially CNN in the field of computer vision to study and visualize the features learned by neurons. Recent studies have investigated visualization of RNN and its variants. Tang et al. (2017) visualized the memory vectors to understand the behavior of LSTM and gated recurrent unit (GRU) in speech recognition task. For given words in a sentence, Li et al. (2016) employed heat maps to study sensitivity and meaning composition in recurrent networks. Ming et al. (2017) proposed a tool, RNNVis to visualize hidden states based on RNN's expected response to

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 154–164
Brussels, Belgium, November 1, 2018. ©2018 Association for Computational Linguistics

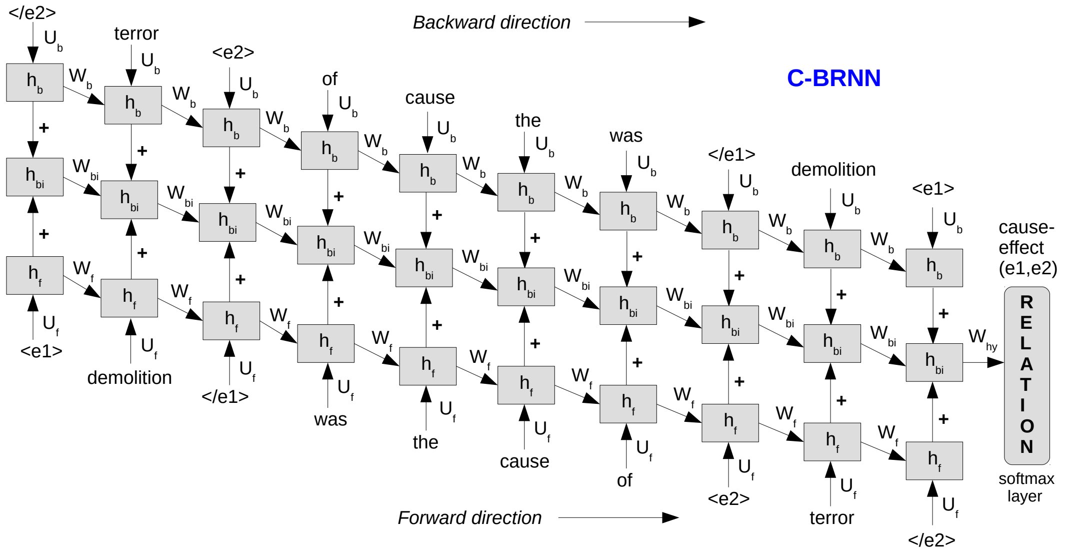


Figure 1: Connectionist Bi-directional Recurrent Neural Network (C-BRNN) (Vu et al., 2016a)

inputs. Peters et al. (2018) studied the internal states of deep bidirectional language model to learn contextualized word representations and observed that the higher-level hidden states capture word semantics, while lower-level states capture syntactical aspects. Despite the possibility of visualizing hidden state activations and performance-based analysis, there still remains a challenge for humans to interpret hidden behavior of the“black box” networks that raised questions in the NLP community as to verify that the network behaves as expected. In this aspect, we address the cumulative nature of RNN with the text input and computed response to answer “how does it aggregate and build the semantic meaning of a sentence word by word at each time point in the sequence for each category in the data”.

Contribution: In this work, we analyze and interpret the cumulative nature of RNN via a proposed technique named as *Layer-wIse-Semantic-Accumulation* (LISA) for explaining decisions and detecting the most likely (i.e., saliency) patterns that the network relies on while decision making. We demonstrate (1) *LISA*: “How an RNN accumulates or builds semantics during its sequential processing for a given text example and expected response” (2) *Example2pattern*: “How the saliency patterns look like for each category in the data according to the network in decision making”. We analyse the sensitiveness of RNNs about different inputs to check the increase or decrease in prediction scores. For an example sentence that is classified correctly, we identify and extract a saliency pattern (N-grams of words in order learned by the network) that contributes the most in prediction score. Therefore, the term *example2pattern* transformation for each category in the data. We employ two relation classification datasets: SemEval 10 Task 8 and TAC KBP Slot Filling (SF) Shared Task (ST) to explain RNN predictions via the proposed *LISA* and *example2pattern* techniques.

2 Connectionist Bi-directional RNN

We adopt the bi-directional recurrent neural network architecture with ranking loss, proposed by Vu et al. (2016a). The network consists of three parts: a forward pass which processes the original sentence word by word (Equation 1); a backward pass which processes the reversed sentence word by word (Equation 2); and a combination of both (Equation 3). The forward and backward passes are combined by adding their hidden layers. There is also a connection to the previous combined hidden layer with weight W_{bi} with a motivation to include all intermediate hidden layers into the final decision of the network (see Equation 3). They named the neural architecture as ‘Connectionist Bi-directional RNN’ (C-BRNN). Figure 1 shows the C-BRNN architecture, where all the three parts are trained jointly.

$$h_{f_t} = f(U_f \cdot w_t + W_f \cdot h_{f_{t-1}}) \tag{1}$$

$$h_{b_t} = f(U_b \cdot w_{n-t+1} + W_b \cdot h_{b_{t+1}}) \tag{2}$$

$$h_{bi_t} = f(h_{f_t} + h_{b_t} + W_{bi} \cdot h_{bi_{t-1}}) \tag{3}$$

where w_t is the word vector of dimension d for a word at time step t in a sentence of length n.

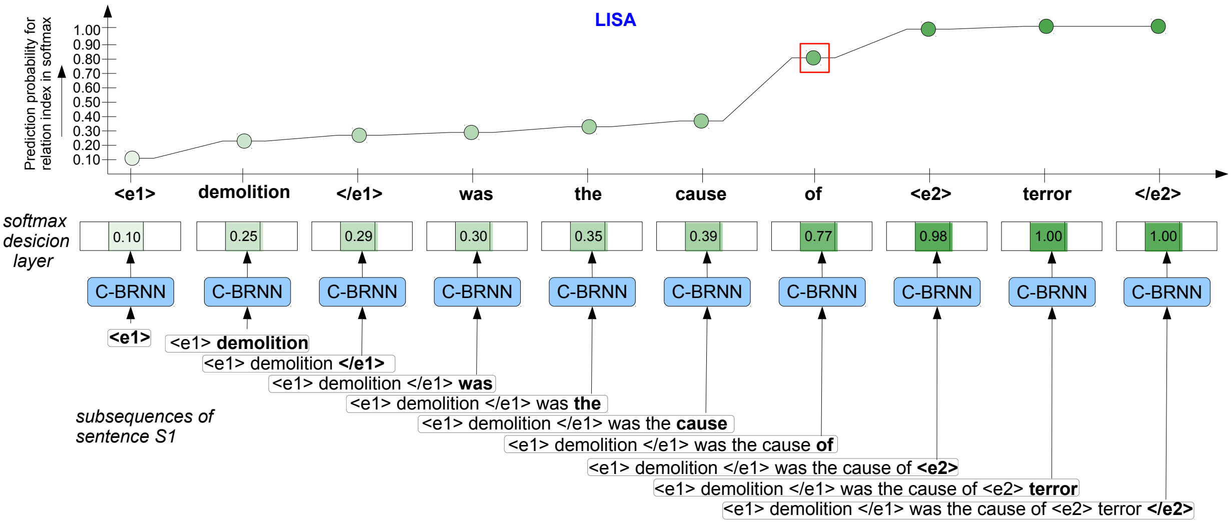

LISA
Prediction probability for relation index in softmax
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
<e1>
demolition
</e1>
was
the
cause
of
<e2>
terror
</e2>
softmax desicion layer
0.10
0.25
0.29
0.30
0.35
0.39
0.77
0.98
1.00
1.00
C-BRNN
subsequences of sentence S1
<e1>
<e1> demolition
<e1> demolition </e1>
<e1> demolition </e1> was
<e1> demolition </e1> was the
<e1> demolition </e1> was the cause
<e1> demolition </e1> was the cause of
<e1> demolition </e1> was the cause of <e2>
<e1> demolition </e1> was the cause of <e2> terror
<e1> demolition </e1> was the cause of <e2> terror </e2>

Figure 2: An illustration of Layer-wIse Semantic Accumulation (LISA) in C-BRNN, where we compute prediction score for a (known) relation type at each of the input subsequence. The highlighted indices in the softmax layer signify one of the relation types, i.e., *cause*-effect(e1, e2) in SemEval10 Task 8 dataset. The bold signifies the last word in the subsequence. Note: Each word is represented by N-gram (N=3, 5 or 7), therefore each input subsequence is a sequence of N-grams. E.g., the word 'of' → 'cause of <e2>' for N=3. To avoid complexity in this illustration, each word is shown as a uni-gram.

D is the hidden unit dimension. $U_f \in \mathbb{R}^{d \times D}$ and $U_b \in \mathbb{R}^{d \times D}$ are the weight matrices between hidden units and input w_t in forward and backward networks, respectively; $W_f \in \mathbb{R}^{D \times D}$ and $W_b \in \mathbb{R}^{D \times D}$ are the weights matrices connecting hidden units in forward and backward networks, respectively. $W_{bi} \in \mathbb{R}^{D \times D}$ is the weight matrix connecting the hidden vectors of the combined forward and backward network. Following Gupta et al. (2015) during model training, we use 3-gram and 5-gram representation of each word w_t at timestep t in the word sequence, where a 3-gram for w_t is obtained by concatenating the corresponding word embeddings, i.e., $w_{t-1}w_tw_{t+1}$.

Ranking Objective: Similar to Santos et al. (2015) and Vu et al. (2016a), we applied the ranking loss function to train C-BRNN. The ranking scheme offers to maximize the distance between the true label y^+ and the best competitive label c^- given a data point x. It is defined as-

$$\begin{aligned}\mathcal{L} &= \log(1 + \exp(\gamma(m^+ - s_\theta(x)_{y^+}))) \\ &\quad + \log(1 + \exp(\gamma(m^- + s_\theta(x)_{c^-})))\end{aligned} \tag{4}$$

where $s_\theta(x)_{y^+}$ and $s_\theta(x)_{c^-}$ being the scores for the classes y^+ and c^-, respectively. The parameter γ controls the penalization of the prediction errors and m^+ and m are margins for the correct and incorrect classes. Following Vu et al. (2016a), we set γ = 2, m^+ = 2.5 and m^- = 0.5.

Model Training and Features: We represent each word by the concatenation of its word embedding and position feature vectors. We use word2vec (Mikolov et al., 2013) embeddings, that are updated during model training. As position features in relation classification experiments, we use position indicators (PI) (Zhang and Wang, 2015) in C-BRNN to annotate target entity/nominals in the word sequence, without necessity to change the input vectors, while it increases the length of the input word sequences, as four independent words, as position indicators (<e1>, </ e1>, <e2>, </e2>) around the relation arguments are introduced.

In our analysis and interpretation of recurrent neural networks, we use the trained C-BRNN (Figure 1) (Vu et al., 2016a) model.

3 LISA and Example2Pattern in RNN

There are several aspects in interpreting the neural network, for instance via (1) *Data*: "Which dimensions of the data are the most relevant for the task" (2) *Prediction* or *Decision*: "Explain why a certain pattern" is classified in a certain way (3) *Model*: "How patterns belonging to each category in the data look like according to the network".

In this work, we focus to explain RNN via *decision* and *model* aspects by finding the patterns that explains "why" a model arrives at a particular decision for each category in the data and verifies that model behaves as expected. To do so, we propose a technique named as LISA that interprets RNN about "how it accumulates and builds meaningful semantics of a sentence word by word" and "how the saliency patterns look like according to the network" for each category in the data while decision making. We extract the saliency patterns via *example2pattern* transformation.

LISA Formulation: To explain the cumulative nature of recurrent neural networks, we show how does it build semantic meaning of a sentence word by word belonging to a particular category in the data and compute prediction scores for the expected category on different inputs, as shown in Figure 2. The scheme also depicts the contribution of each word in the sequence towards the final classification score (prediction probability).

At first, we compute different subsequences of word(s) for a given sequence of words (i.e., sentence). Consider a sequence $\mathbf{S}$ of words $[w_1, w_2, ..., w_k, ..., w_n]$ for a given sentence S of length n. We compute n number of subsequences, where each subsequence $\mathbf{S}_{\leq k}$ is a subvector of words $[w_1, ...w_k]$, i.e., $\mathbf{S}_{\leq k}$ consists of words preceding and including the word w_k in the sequence $\mathbf{S}$. In context of this work, extending a subsequence by a word means appending the subsequence by the next word in the sequence. Observe that the number of subsequences, n is equal to the total number of time steps in the C-BRNN.

Next is to compute RNN prediction score for the category R associated with sentence S. We compute the score via the autoregressive conditional $P(R|\mathbf{S}_{\leq k}, \mathbb{M})$ for each subsequence $\mathbf{S}_{\leq k}$, as-

$$P(R|\mathbf{S}_{\leq k}, \mathbb{M}) = softmax(W_{hy} \cdot h_{bi_k} + b_y) \quad (5)$$

using the trained C-BRNN (Figure 1) model $\mathbb{M}$. For each $k \in [1, n]$, we compute the network prediction, $P(R|\mathbf{S}_{\leq k}, \mathbb{M})$ to demonstrate the cumulative property of recurrent neural network that builds meaningful semantics of the sequence $\mathbf{S}$ by extending each subsequence $\mathbf{S}_{\leq k}$ word by word. The internal state h_{bi_k} (attached to softmax layer as in Figure 1) is involved in decision making for each input subsequence $\mathbf{S}_{\leq k}$ with bias vector $b_y \in \mathbb{R}^C$ and hidden-to-softmax weights matrix $W_{hy} \in \mathbb{R}^{D \times C}$ for C categories.

The *LISA* is illustrated in Figure 2, where each word in the sequence contributes to final classification score. It allows us to understand the network decisions via peaks in the prediction score over different subsequences. The peaks signify the saliency patterns (i.e., sequence of words) that the network has learned in order to make decision. For instance, the input word '*of*' following the subsequence '*<e1> demolition </e1> was the cause*' introduces a sudden increase in prediction score for the relation type *cause-effect*(e1, e2). It suggests that the C-BRNN collects the semantics layer-wise via temporally organized subsequences. Observe that the subsequence '*...cause of*' is salient enough in decision making (i.e., prediction score=0.77), where the next subsequence '*...cause of <e2>*' adds in the score to get 0.98.

Algorithm 1 Example2pattern Transformation

```
Input: sentence $S$, length $n$, category $R$, threshold $\tau$, C-BRNN $\mathbb{M}$, N-gram size N
Output: N-gram saliency pattern $patt$
for $k$ in 1 to $n$ do
    compute N-gram$_k$ (eqn 8) of words in $S$
for $k$ in 1 to $n$ do
    compute $\mathbf{S}_{\leq k}$ (eqn 7) of N-grams
    compute $P(R|\mathbf{S}_{\leq k}, \mathbb{M})$ using eqn 5
    if $P(R|\mathbf{S}_{\leq k}, \mathbb{M}) \geq \tau$ then
        return $patt \leftarrow \mathbf{S}_{\leq k}[-1]$
```

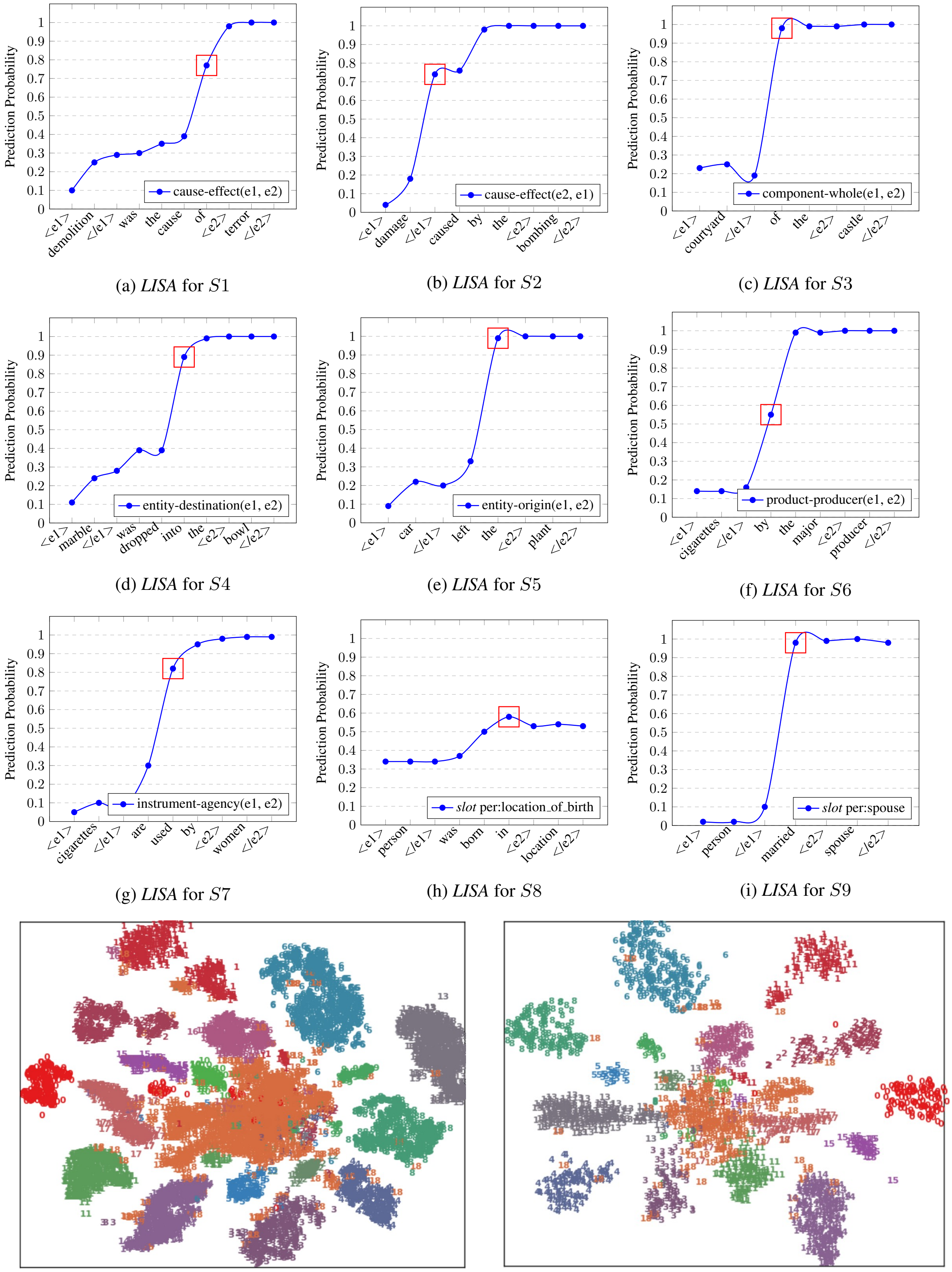
Example2pattern for Saliency Pattern: To further interpret RNN, we seek to identify and extract the most likely input pattern (or phrases) for a given class that is discriminating enough in decision making. Therefore, each example input is transformed into a saliency pattern that informs us about the network learning. To do so, we first compute N-gram for each word w_t in the sentence S. For instance, a 3-gram representation of w_t is given by w_{t-1}, w_t, w_{t+1}. Therefore, an N-gram (for N=3) sequence $\mathbf{S}$ of words is represented as $[[w_{t-1}, w_t, w_{t+1}]_{t=1}^{n}]$, where w_0 and w_{n+1} are PADDING (zero) vectors of embedding dimension.

Following Vu et al. (2016a), we use N-grams (e.g., tri-grams) representation for each word in each subsequence $\mathbf{S}_{\leq k}$ that is input to C-BRNN to compute $P(R|\mathbf{S}_{\leq k})$, where the N-gram (N=3) subsequence $\mathbf{S}_{\leq k}$ is given by,

$$\begin{aligned}\mathbf{S}_{\leq k} = [[PADDING, w_1, w_2]_1, [w_1, w_2, w_3]_2, ...,\\ [w_{t-1}, w_t, w_{t+1}]_t, ..., [w_{k-1}, w_k, w_{k+1}]_k]\end{aligned} \quad (6)$$

$$\mathbf{S}_{\leq k} = [tri_1, tri_2, ..., tri_t, ...tri_k] \quad (7)$$

for $k \in [1, n]$. Observe that the 3-gram tri_k con-



(a) *LISA* for $S1$

(b) *LISA* for $S2$

(c) *LISA* for $S3$

(d) *LISA* for $S4$

(e) *LISA* for $S5$

(f) *LISA* for $S6$

(g) *LISA* for $S7$

(h) *LISA* for $S8$

(i) *LISA* for $S9$

(j) t-SNE Visualization for training set

(k) t-SNE Visualization for testing set

Figure 3: (a-i) Layer-wIse Semantic Accumulation (LISA) by C-BRNN for different relation types in SemEval10 Task 8 and TAC KBP Slot Filling datasets. The square in red color signifies that the relation is correctly detected with the input subsequence (enough in decision making). (j-k) t-SNE visualization of the last combined hidden unit (h_{bi}) of C-BRNN computed using the SemEval10 train and test sets.

ID	Relation/Slot Types	Example Sentences	Example2Pattern
$S1$	cause-effect(e1, e2)	<e1> demolition </e1> was the cause of <e2> terror </e2>	cause of <e2>
$S2$	cause-effect(e2, e1)	<e1> damage </e1> caused by the <e2> bombing </e2>	damage </e1> caused
$S3$	component-whole(e1, e2)	<e1> countyard </e1> of the <e2> castle </e2>	</e1> of the
$S4$	entity-destination(e1,e2)	<e1> marble </e1> was dropped into the <e2> bowl </e2>	dropped into the
$S5$	entity-origin(e1, e2)	<e1> car </e1> left the <e2> plant </e2>	left the <e2>
$S6$	product-produce(e1, e2)	<e1> cigarettes </e1> by the major <e2> producer </e2>	</e1> by the
$S7$	instrument-agency(e1, e2)	<e1> cigarettes </e1> are used by <e2> women </e2>	</e1> are used
$S8$	per:loc_of_birth(e1, e2)	<e1> person </e1> was born in <e2> location </e2>	born in <e2>
$S9$	per:spouse(e1, e2)	<e1> person </e1> married <e2> spouse </e2>	</e1> married <e2>

Table 1: Example Sentences for *LISA* and *example2pattern* illustrations. The sentences $S1$-$S7$ belong to SemEval10 Task 8 dataset and $S8$-$S9$ to TAC KBP Slot Filling (SF) shared task dataset.

sists of the word w_{k+1}, if k $\neq$ n. To generalize for $i \in [1, \lfloor N/2 \rfloor]$, an N-gram$_k$ of size N for word w_k in C-BRNN is given by-

$$\text{N-gram}_k = [w_{k-i}, ..., w_k, ..., w_{k+i}]_k \qquad (8)$$

Algorithm 1 shows the transformation of an example sentence into pattern that is salient in decision making. For a given example sentence S with its length n and category R, we extract the most salient N-gram (N=3, 5 or 7) pattern $patt$ (the last N-gram in the N-gram subsequence $\mathbf{S}_{\leq k}$) that contributes the most in detecting the relation type R. The threshold parameter τ signifies the probability of prediction for the category R by the model $\mathbb{M}$. For an input N-gram sequence $\mathbf{S}_{\leq k}$ of sentence S, we extract the last N-gram, e.g., tri_k that detects the relation R with prediction score above τ. By manual inspection of patterns extracted at different values (0.4, 0.5, 0.6, 0.7) of τ, we found that $\tau = 0.5$ generates the most salient and interpretable patterns. The saliency pattern detection follows LISA as demonstrated in Figure 2, except that we use N-gram (N =3, 5 or 7) input to detect and extract the key relationship patterns.

4 Analysis: Relation Classification

Given a sentence and two annotated nominals, the task of binary relation classification is to predict the semantic relations between the pairs of nominals. In most cases, the context in between the two nominals define the relationship. However, Vu et al. (2016a) has shown that the extended context helps. In this work, we focus on the building semantics for a given sentence using relationship contexts between the two nominals.

We analyse RNNs for *LISA* and *example2pattern* using two relation classification datsets: (1) SemEval10 Shared Task 8 (Hendrickx et al., 2009) (2) TAC KBP Slot Filling (SF) shared task[1] (Adel and Schütze, 2015). We demonstrate the sensitiveness of RNN for different subsequences (Figure 2), input in the same order as in the original sentence. We explain its predictions (or judgments) and extract the salient relationship patterns learned for each category in the two datasets.

Input word sequence to C-BRNN	*pp*
<**e1**>	0.10
<e1> **demolition**	0.25
<e1> demolition <**/e1**>	0.29
<e1> demolition </e1> **was**	0.30
<e1> demolition </e1> was **the**	0.35
<e1> demolition </e1> was the **cause**	0.39
<e1> demolition </e1> was the cause **of**	$\underline{0.77}$
<e1> demolition </e1> was the cause of <**e2**>	0.98
<e1> demolition </e1> was the cause of <e2> **terror**	1.00
<e1> demolition </e1> was the cause of <e2> terror <**/e2**>	1.00

Table 2: Semantic accumulation and sensitivity of C-BRNN over subsequences for sentence $S1$. Bold indicates the last word in the subsequence. *pp*: prediction probability in the softmax layer for the relation type. The underline signifies that the *pp* is sufficient enough (τ=0.50) in detecting the relation. Saliency patterns, i.e., N-grams can be extracted from the input subsequence that leads to a sudden peak in *pp*, where *pp* $\geq \tau$.

4.1 SemEval10 Shared Task 8 dataset

The relation classification dataset of the Semantic Evaluation 2010 (SemEval10) shared task 8 (Hendrickx et al., 2009) consists of 19 relations (9 directed relations and one artificial class `Other`), 8,000 training and 2,717 testing sentences. We split the training data into train (6.5k) and development (1.5k) sentences to optimize the C-BRNN

[1] data from the slot filler classification component of the slot filling pipeline, treated as relation classification

Relation	3-gram Patterns	5-gram Patterns	7-gram Patterns
cause-effect(e1,e2)	</e1> cause <e2>	the leading causes of <e2>	is one of the leading causes of
	</e1> caused a	the main causes of <e2>	is one of the main causes of
	that cause respiratory	</e1> leads to <e2> inspiration	</e1> that results in <e2> hardening </e2>
	which cause acne	</e1> that results in <e2>	</e1> resulted in the <e2> loss </e2>
	leading causes of	</e1> resulted in the <e2>	<e1> sadness </e1> leads to <e2> inspiration
cause-effect(e2,e1)	caused due to	</e1> has been caused by	</e1> is caused by a <e2> comet
	comes from the	</e1> are caused by the	</e1> however has been caused by the
	arose from an	</e1> arose from an <e2>	</e1> that has been caused by the
	caused by the	</e1> caused due to <e2>	that has been caused by the <e2>
	radiated from a	infection </e2> results in an	<e1> product </e1> arose from an <e2>
content-container(e1,e2)	in a <e2>	</e1> was contained in a	</e1> was contained in a <e2> box
	was inside a	</e1> was discovered inside a	</e1> was in a <e2> suitcase </e2>
	contained in a	</e1> were in a <e2>	</e1> were in a <e2> box </e2>
	hidden in a	is hidden in a <e2>	</e1> was inside a <e2> box </e2>
	stored in a	</e1> was contained in a	</e1> was hidden in an <e2> envelope
product-produce(e1,e2)	</e1> released by	</e1> issued by the <e2>	<e1> products </e1> created by an <e2>
	</e1> issued by	</e1> was prepared by <e2>	</e1> by an <e2> artist </e2> who
	</e1> created by	was written by a <e2>	</e1> written by most of the <e2>
	by the <e2>	</e1> built by the <e2>	temple </e1> has been built by <e2>
	of the <e1>	</e1> are made by <e2>	</e1> were founded by the <e2> potter
whole(e1, e2) component-	</e1> of the	</e1> of the <e2> device	the <e1> timer </e1> of the <e2>
	of the <e2>	</e1> was a part of	</e1> was a part of the romulan
	part of the	</e1> is part of the	</e1> was the best part of the
	</e1> of <e2>	is a basic element of	</e1> is a basic element of the
	</e1> on a	</e1> is part of a	are core components of the <e2> solutions
entity-destination(e1,e2)	put into a	have been moving into the	</e1> have been moving back into <e2>
	released into the	was dropped into the <e2>	</e1> have been moving into the <e2>
	</e1> into the	</e1> moved into the <e2>	</e1> have been dropped into the <e2>
	moved into the	were released into the <e2>	</e1> have been released back into the
	added to the	</e1> have been exported to	power </e1> is exported to the <e2>
instrument-agency(e1,e2)	</e1> are used	</e1> assists the <e2> eye	cigarettes </e1> are used by <e2> women
	used by <e2>	</e1> are used by <e2>	<e1> telescope </e1> assists the <e2> eye
	</e1> is used	</e1> were used by some	<e1> practices </e1> for <e2> engineers </e2>
	set by the	</e1> with which the <e2>	the best <e1> tools </e1> for <e2>
	</e1> set by	readily associated with the <e2>	<e1> wire </e1> with which the <e2>

Table 3: SemEval10 Task 8 dataset: N-Gram (3, 5 and 7) saliency patterns extracted for different relation types by C-BRNN with PI

network. For instance, an example sentence with relation label is given by-

`The <e1> demolition </e1> was the cause of <e2> terror </e2> and communal divide is just a way of not letting truth prevail.` → `cause-effect(e1,e2)`

The terms `demolition` and `terror` are the relation arguments or nominals, where the phrase `was the cause of` is the relationship context between the two arguments. Table 1 shows the examples sentences (shortened to argument1+relationship context+argument2) drawn from the development and test sets that we employed to analyse the C-BRNN for semantic accumulation in our experiments. We use the similar experimental setup as Vu et al. (2016a).

LISA Analysis: As discussed in Section 3, we interpret C-BRNN by explaining its predictions via the semantic accumulation over the subsequences $\mathbf{S}_{\leq k}$ (Figure 2) for each sentence S. We select the example sentences $S1$-$S7$ (Table 1) for which the network predicts the correct relation type with high scores. For an example sentence $S1$, Table 2 illustrates how different subsequences are input to C-BRNN in order to compute prediction scores pp in the softmax layer for the relation `cause-effect(e1, e2)`. We use tri-gram (section 3) word representation for each word for the examples $S1$-$S7$.

Figures 3a, 3b, 3c, 3d 3e, 3f and 3g demonstrate the cumulative nature and sensitiveness of RNN via prediction probability (pp) about different inputs for sentences $S1$-$S7$, respectively. For

Slots	N-gram Patterns
per-spouse(e1,e2)	</e1> wife of
	</e1> , wife
	</e1> wife
	</e1> married <e2>
	</e1> marriages to
per-location_of_birth(e1,e2)	was born in
	born in <e2>
	a native of
	</e1> from <e2>
	</e1> 's hometown

Table 4: TAC KBP SF dataset: Tri-gram saliency patterns extracted for slots *per:spouse*(e1, e2) and *per:location_of_birth*(e1,e2)

instance in Figure 3a and Table 2, the C-BRNN builds meaning of the sentence $S1$ word by word, where a sudden increase in pp is observed when the input subsequence `<e1> demolition </e1> was the cause` is extended with the next term `of` in the word sequence **S**. Note that the relationship context between the arguments `demolition` and `terror` is sufficient enough in detecting the relationship type. Interestingly, we also observe that the prepositions (such as `of`, `by`, `into`, etc.) in combination with verbs are key features in building the meaningful semantics.

Saliency Patterns via example2pattern Transformation: Following the discussion in Section 3 and Algorithm 1, we transform each correctly identified example into pattern by extracting the most likely N-gram in the input subsequence(s). In each of the Figures 3a, 3b, 3c, 3d 3e, 3f and 3g, the square box in red color signifies that the relation type is correctly identified (when $\tau = 0.5$) at this particular subsequence input (without the remaining context in the sentence). We extract the last N-gram of such a subsequence.

Table 1 shows the *example2pattern* transformations for sentences $S1$-$S7$ in SemEval10 dataset, derived from Figures 3a-3g, respectively with N=3 (in the N-grams). Similarly, we extract the salient patterns (3-gram, 5-gram and 7-gram) (Table 3) for different relationships. We also observe that the relation types `content-container(e1, e2)` and `instrument-agency(e1, e2)` are mostly defined by smaller relationship contexts (e.g, 3-gram), however `entity-destination(e1,e2)` by larger contexts (7-gram).

4.2 TAC KBP Slot Filling dataset

We investigate another dataset from TAC KBP Slot Filling (SF) shared task (Surdeanu, 2013), where we use the relation classification dataset by Adel et al. (2016) in the context of slot filling. We have selected the two slots: *per:loc_of_birth* and *per:spouse* out of 24 types.

LISA Analysis: Following Section 4.1, we analyse the C-BRNN for LISA using sentences $S8$ and $S9$ (Table 1). Figures 3h and 3i demonstrate the cumulative nature of recurrent neural network, where we observe that the salient patterns `born in <e2>` and `</e1> married` $e2$ lead to correct decision making for $S8$ and $S9$, respectively. Interestingly for $S8$, we see a decrease in prediction score from 0.59 to 0.52 on including terms in the subsequence, following the term `in`.

Saliency Patterns via example2pattern Transformation: Following Section 3 and Algorithm 1, we demonstrate the *example2pattern* transformation of sentences $S8$ and $S9$ in Table 1 with tirgrams. In addition, Table 4 shows the tri-gram salient patterns extracted for the two slots.

5 Visualizing Latent Semantics

In this section, we attempt to visualize the hidden state of each test (and train) example that has accumulated (or built) the meaningful semantics during sequential processing in C-BRNN. To do this, we compute the last hidden vector h_{bi} of the combined network (e.g., h_{bi} attached to the softmax layer in Figure 1) for each test (and train) example and visualize (Figure 3k and 3j) using t-SNE (Maaten and Hinton, 2008). Each color represents a relation-type. Observe the distinctive clusters of accumulated semantics in hidden states for each category in the data (SemEval10 Task 8).

6 Conclusion and Future Work

We have demonstrated the cumulative nature of recurrent neural networks via sensitivity analysis over different inputs, i.e., *LISA* to understand how they build meaningful semantics and explain predictions for each category in the data. We have also detected a salient pattern in each of the example sentences, i.e., *example2pattern transformation* that the network learns in decision making. We extract the salient patterns for different categories in two relation classification datasets.

In future work, it would be interesting to analyse the sensitiveness of RNNs with corruption in the salient patterns. One could also investigate visualizing the dimensions of hidden states (activation maximization) and word embedding vectors with the network decisions over time. We forsee to apply *LISA* and *example2pattern* on different tasks such as document categorization, sentiment analysis, language modeling, etc. Another interesting direction would be to analyze the bag-of-word neural topic models such as DocNADE (Larochelle and Lauly, 2012) and iDocNADE (Gupta et al., 2018b) to interpret their semantic accumulation during autoregressive computations in building document representation(s). We extract the saliency patterns for each category in the data that can be effectively used in instantiating pattern-based information extraction systems, such as bootstrapping entity (Gupta and Manning, 2014) and relation extractors (Gupta et al., 2018e).

Acknowledgments

We thank Heike Adel for providing us with the TAC KBP dataset used in our experiments. We express appreciation for our colleagues Bernt Andrassy, Florian Buettner, Ulli Waltinger, Mark Buckley, Stefan Langer, Subbu Rajaram, Yatin Chaudhary, and anonymous reviewers for their in-depth review comments. This research was supported by Bundeswirtschaftsministerium (`bmwi.de`), grant 01MD15010A (Smart Data Web) at Siemens AG- CT Machine Intelligence, Munich Germany.

References

Heike Adel, Benjamin Roth, and Hinrich Schütze. 2016. Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838. Association for Computational Linguistics.

Heike Adel and Hinrich Schütze. 2015. Cis at tac cold start 2015: Neural networks and coreference resolution for slot filling. *Proc. TAC2015*.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Alexey Dosovitskiy and Thomas Brox. 2016. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4829–4837.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Pankaj Gupta, Bernt Andrassy, and Hinrich Schütze. 2018a. Replicated siamese lstm in ticketing system for similarity learning and retrieval in asymmetric texts. In *Proceedings of the Workshop on Semantic Deep Learning (SemDeep-3) in the 27th International Conference on Computational Linguistics (COLING2018)*. The COLING 2018 organizing committee.

Pankaj Gupta, Florian Buettner, and Hinrich Schütze. 2018b. Document informed neural autoregressive topic models. Researchgate preprint doi: 10.13140/RG.2.2.12322.73925.

Pankaj Gupta, Subburam Rajaram, Thomas Runkler, Hinrich Schütze, and Bernt Andrassy. 2018c. Neural relation extraction within and across sentence boundaries. Researchgate preprint doi: 10.13140/RG.2.2.16517.04327.

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Bernt Andrassy. 2018d. Deep temporal-recurrent-replicated-softmax for topical trends over time. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1079–1089, New Orleans, USA. Association of Computational Linguistics.

Pankaj Gupta, Benjamin Roth, and Hinrich Schütze. 2018e. Joint bootstrapping machines for high confidence relation extraction. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers)*, pages 26–36, New Orleans, USA. Association of Computational Linguistics.

Pankaj Gupta, Thomas Runkler, Heike Adel, Bernt Andrassy, Hans-Georg Zimmermann, and Hinrich Schütze. 2015. Deep learning methods for the extraction of relations in natural language text. Technical report, Technical University of Munich, Germany.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547. The COLING 2016 Organizing Committee.

Sonal Gupta and Christopher Manning. 2014. Spied: Stanford pattern based information extraction and diagnostics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 38–44.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Geoffrey E Hinton. 2012. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2708–2716. Curran Associates, Inc.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the Workshop at ICLR*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu. 2017. Understanding hidden memories of recurrent neural networks. *arXiv preprint arXiv:1710.10777*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116. Association for Computational Linguistics.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2017. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, volume 16, pages 2786–2792.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL*. Association for Computational Linguistics.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *TAC*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432. Association for Computational Linguistics.

Zhiyuan Tang, Ying Shi, Dong Wang, Yang Feng, and Shiyue Zhang. 2017. Memory visualization for gated recurrent neural networks in speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 2736–2740. IEEE.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016a. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539, San Diego, California USA. Association for Computational Linguistics.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016b. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *Proceedings of the Acoustics, Speech and Signal Processing (ICASSP)*, pages 6060–6064, Shanghai, China. IEEE.

Yequan Wang, Minlie Huang, xiaoyan zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615. Association for Computational Linguistics.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network.