Investigating how well contextual features are captured by bi-directional recurrent neural network models

Kushal Chawla[1*], **Sunil Kumar Sahu**[2*], **Ashish Anand**[3]
[1]Adobe Research, Big Data Experience Lab, Bangalore, Karnataka, India
[2]National Center for Text Mining, The University of Manchester, United Kingdom
[3]Department of Computer Science and Engineering, IIT Guwahati, Assam, India
kchawla@adobe.com
sunil.sahu@manchester.ac.uk
anand.ashish@iitg.ernet.in

Abstract

Learning algorithms for natural language processing (NLP) tasks traditionally rely on manually defined relevant contextual features. On the other hand, neural network models using an only distributional representation of words have been successfully applied for several NLP tasks. Such models learn features automatically and avoid explicit feature engineering. Across several domains, neural models become a natural choice specifically when limited characteristics of data are known. However, this flexibility comes at the cost of interpretability. In this paper, we define three different methods to investigate ability of bi-directional recurrent neural networks (RNNs) in capturing contextual features. In particular, we analyze RNNs for sequence tagging tasks. We perform a comprehensive analysis on general as well as biomedical domain datasets. Our experiments focus on important contextual words as features, which can easily be extended to analyze various other feature types. We also investigate positional effects of context words and show how the developed methods can be used for error analysis.

1 Introduction

Learning approaches for NLP tasks can be broadly put into two categories based on the way features are obtained or defined. The traditional way is to design features according to a specific problem setting and then use appropriate learning approach. Examples of such methods include classification algorithms like SVM (Hong, 2005) and CRF (Lafferty et al., 2001) among others for several NLP tasks. A significant proportion of overall effort is spent on feature engineering itself. The desire to obtain better performance on a particular problem makes the researchers come up with a domain and task-specific set of features. The primary advantage of using these models is their interpretability. However, dependence on handcrafted features limits their applicability in low resource domain where obtaining a rich set of features is difficult.

On the other hand, neural network models provide a more generalised way of approaching problems in NLP domain. The models can learn relevant features with minimal efforts in explicit feature engineering. This ability allows the use of such models for problems in low resource domain.

The primary drawback of neural network models is that they are too complicated to interpret as the features are not manually defined. Neural networks have been applied significantly to various tasks without many insights on what the underlying structural properties are and how the models learn to classify the inputs correctly. Mostly inspired by computer vision (Simonyan et al., 2013; Nguyen et al., 2015), several mathematical and visual techniques have been developed in this direction (Elman, 1989; Karpathy et al., 2015; Li et al., 2016).

In contrast to the existing works, this study aims to investigate ability of recurrent neural models to capture important context words. Towards this goal, we define multiple measures based on word erasure technique (Li et al., 2016). We do a comprehensive analysis of performance of bi-directional recurrent neural network models for sequence tagging tasks using these measures.

*Part of this work was done while authors were students at IIT Guwahati.

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 273–282

Analysis is focused at understanding how well the relevant contextual words are being captured by different neural models in different settings. The analysis provides a general tool to compare between different models, show that how neural networks follow our intuition by giving importance to more relevant words, study positional effects of context words and provide error analysis for improving the results.

2 Proposed Methods

A sequence tagging task involves assigning a tag (from a predefined set) to each element present in a given sequence. We model Name Entity Recognition (NER) as a sequence tagging task. We follow BIO-tagging scheme, where each named entity type is associated with two labels, $B-entity$ (standing for *Beginning*) and $I-entity$ (standing for *Intermediate*). The BIO scheme uses another label O(standing for *Other*) for all the context or non-entity words.

In this section, we discuss three methods to calculate the importance score of context words. Each method creates a different ranking of context words corresponding to each entity type for a given dataset. The methods range from simple frequency based to considering sentence level or individual word level effects. We assume that we have a pretrained model M on a given dataset.

2.1 Based on word frequency

For a given sentence $S \in$ test set D, consider a window of a particular size around each entity phrase (single or multi word, defined by true tags) w_e in S. We increment the score (corresponding to w_e's entity type e only) for each of the context words present in this window by one. For instance, the CoNLL-2003 shared task data (described in section 3.2) has 4 entity types, namely, *organization (ORG)*, *location (LOC)*, *person (PER)* and *miscellaneous ($MISC$)*. The corresponding labels under BIO-tagging scheme are *B-ORG*, *I-ORG*, *B-LOC*, *I-LOC* and so on. For a 2-word phrase with true tags as (*B-LOC*, *I-LOC*), the score corresponding to LOC for each context word (with true tag as O) in the window is incremented by one. Let the score for a context word w_c corresponding to entity type e in one sentence be $A(w_c, e, S)$.

Hence the relevance score is calculated as follows:

$$I(w_c, e) = \frac{\sum\limits_{\forall S \in D} A(w_c, e, S)}{\sum\limits_{\forall w_c} \sum\limits_{\forall S \in D} A(w_c, e, S)} \quad (1)$$

Using inverse frequency to account for irrelevant, too frequent words, the score can be calculated as follows:

$$I(w_c, e) \quad (2)$$
$$= \left(\frac{\sum\limits_{\forall S \in D} A(w_c, e, S)}{\sum\limits_{\forall w_c} \sum\limits_{\forall S \in D} A(w_c, e, S)} \right) \left(\frac{\sum\limits_{\forall e'} \sum\limits_{\forall w_c} \sum\limits_{\forall S \in D} A(w_c, e', S)}{\sum\limits_{\forall e'} \sum\limits_{\forall S \in D} A(w_c, e', S) + k} \right)$$

where k accounts for 0 counts and sum over e' means summing over all the remaining entity types. In our experiments, we use k=1 and a window size of 11 (5 words on each side). We refer to these methods collectively as M_WF in rest of the paper.

2.2 Using sentence level log likelihood

In the M_WF method, the relevance of each context word is calculated irrespective of its dependence on other words in the sentence. We define another measure using sentence level log likelihood to take into account the dependency between words in a sentence. We refer to this method as M_SLL in rest of the paper.

Let the set of all context words be W and that of all entity types be E. Define $S_{w_c,e}$ as the set of all sentences where both the word $w_c \in W$ and entity type $e \in E$ are present. We say that an entity type e is present in a sentence S, if $\exists$ a word $\in S$ which has it's true tag corresponding to entity type e. Let $F(w_c, e)$ be the size of set $S_{w_c,e}$.

Now, let the true tag sequence for a sentence S be S_{TAGS}. For a context word $w_c \in S$, let $L_1(w_c, S)$ be the negative log likelihood of S_{TAGS} obtained from pretrained model M. Note that since we are working at a sentence level, $L_1(w_c, S)$ will be same for all the context words and entities present in S.

We adapt the erasure method of Li et al. (2016). Here, we replace the representation of word w_c with a random word representation having same number of dimensions and recalculate the negative log likelihood for the true tag sequence S_{TAGS}. Let this value be $L_2(w_c, S)$. Intuitively, if $S \in S_{w_c,e}$ and w_c is relevant for the entity type e, the probability of the true sequence should decrease when the word is removed from the sentence. Correspondingly, it's negative log likelihood value

should increase. Hence, the score $I(w_c, e)$ for a given word corresponding to the entity type can be calculated in the following manner:

$$I(w_c, e) = \frac{1}{F(w_c, e)} \sum_{\forall S \in S_{w_c,e}} \frac{L_2(w_c, S) - L_1(w_c, S)}{L_1(w_c, S)} \quad (3)$$

2.3 Considering left and right word contexts separately

The relevance scoring method M_SLL does not distinguish between words present in the same sentence. The third method, referred to as M_LRC, works at word level and calculates relevance score of each word by distinguishing its presence in the left or right side of the entity word. The measure is defined in a way that it does take into account of dependency between words in the sentence. In a bi-directional setting, the hidden layer representation for any word in a sentence, is a concatenation of two representations - one which combines words to the left, and the other which combines the words to the right.

In the output layer, we combine the weight parameters and the hidden layer representation by a dot product. We divide this dot product in two parts as discussed below. Say the hidden representation is h and weight parameters corresponding to a tag t $\in T$ (set of all possible tags) are represented by p_t. We can write the dot product $p_t^T h$ as a sum of two dot products $p_{t,L}^T h_L$ and $p_{t,R}^T h_R$, representing the contribution from left and right parts separately. In our experiments, we also include the bias term as a weight parameter.

Now, take a sentence S, a context word w_c in S, and an entity word w_e in S with true tag $t \in T$ corresponding to entity type $e \in E$. Define $AvgSum(w_c, w_e, S)$ as follows:

$$AvgSum(w_c, w_e, S) = \frac{\sum_{\forall f \in T - \{t\}} p_{f,K}^T h_K}{\alpha} \quad (4)$$

where α is the size of the set $T - \{t\}$ and K is either L or R depending on whether the word w_c lies to the left or right of w_e respectively. Notice that this sum is over all the false tags in set T for the word w_e.

With the intuition that the important word should have higher dot product corresponding to true tag than to false tags, we define the score $L_1(w_c, w_e, S)$ as follows:

$$L_1(w_c, w_e, S) = \frac{p_{t,K}^T . h_K - AvgSum(w_c, w_e, S)}{AvgSum(w_c, w_e, S)} \quad (5)$$

We again employ word erasure technique and recompute the above score by replacing the representation of word w_c with a random word representation. We call it $L_2(w_c, w_e, S)$. Now, we can compute the final score for this instance $L(w_c, w_e, S)$ as:

$$L(w_c, w_e, S) = \frac{L_1(w_c, w_e, S) - L_2(w_c, w_e, S)}{L_2(w_c, w_e, S)} \quad (6)$$

The relevance score $I(w_c, e)$ is then computed by taking average of $L(w_c, w_e, S)$ over all instances.

3 Experiments

We consider the task of sequence tagging problem for evaluation and analysis of the proposed methods to interpret neural network models. In particular, we choose the three variants of recurrent neural network models for Named Entity Recognition(NER) task.

3.1 Model architecture

The generic RNN model architecture used for this work is given in figure 1.

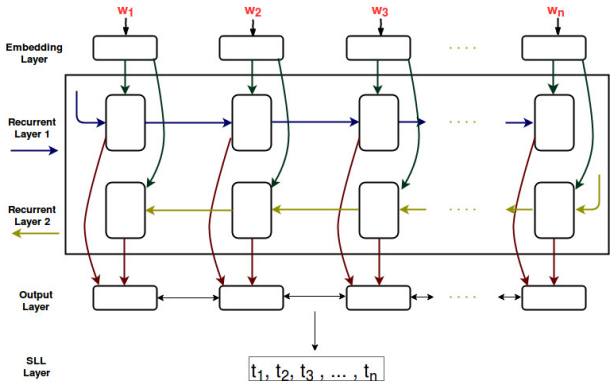


Figure 1: General model architecture for a bi-directional recurrent neural network in sequence tagging problem.

Input layer contains all the words in the sentence. In the embedding layer, each word is represented by it's d dimensional vector representation. The hidden layer contains a bi-directional recurrent neural network which outputs a $2h$ dimensional representation for every word, where h is the number of hidden layer units in the recurrent neural network. In bi-directional models, both the past and future contexts are used to represent the words in a given sentence. Finally, a fully connected network connects the hidden layer to the output layer, which contains scores for each possible tag corresponding to every word in the sen-

tence. A sentence level log likelihood loss function (Collobert et al., 2011) is used in the training process.

For this work, we experiment with standard bi-directional Recurrent Neural Network (Bi-RNN), bi-directional Long Short Term Memory Network (Bi-LSTM) (Graves, 2013; Huang et al., 2015) and bi-directional Gated Recurrent Unit Network(Bi-GRU) (Chung et al., 2014). For simplicity, we refer to these bi-directional models as RNN, LSTM and GRU in rest of the paper.

3.2 Datasets

In this work, we use two NER datasets from diverse domains. One is from generic domain whereas other is from biomedical domain. Statistics of both datasets are given in Table 1.

CoNLL, 2003: This dataset was released as a part of CoNLL-2003 language independent named entity recognition task (Tjong Kim Sang and De Meulder, 2003). Four named entity types have been used: location, person, organization and miscellaneous. For this work, we have used the original split of the English dataset. There were 8 tags used *I-PER*, *B-LOC*, *I-LOC*, *B-ORG*, *I-ORG*, *B-MISC*, *I-MISC* and O. We focus on three entity types, namely, location (LOC), person (PER) and organization (ORG) in our analysis. For this dataset, we use pretrained GloVe 50 dimensional word vectors (Pennington et al., 2014).

JNLPBA, 2004: Released as a part of Bio-Entity recognition task (Kim et al., 2004) at JNLPBA in 2004, this dataset is from GENIA version 3.02 corpus (Kim et al., 2003). There are 5 classes in total - *DNA*, *RNA*, *Cell_line*, *Cell_type* and *Protein*. We use all the classes in our analysis. There are 11 tags, 2 (for begin and intermediate word) for each class and O for other context words. We use 50 dimensional word vectors trained using skip-gram method on a biomedical corpus (Mikolov et al., 2013a; Mikolov et al., 2013b). For this work, we calculate the relevance scores for all the words which have their true tag as O for any test instance in the two datasets.

3.3 Correlation measures

In the output (last) layer we take dot product between weight parameters and the hidden layer outputs and expect that this value (normalized) would be highest corresponding to the true tag. To obtain these similarities between distributions of hidden layer outputs to the weight parameters, we consider two other measures apart from dot product:

1. **Kullback-Leibler Divergence**: Given two discrete probability distributions **A** and **B**, the Kullback-Leibler Divergence(or KL Divergence) from **B** to **A** is computed in the following manner:

$$D_{KL}(A||B) = \sum_i A(i) \log \frac{A(i)}{B(i)} \quad (7)$$

   $D_{KL}(A||B)$ may be interpreted as a measure to see that how good the distribution **B** approximates the distribution **A**. For our experiments, we take normalized weight parameters as **A** and hidden representations as **B**. The lower this KL-divergence is, higher is the correlation between **A** and **B**.

2. **Pearson Correlation Coefficient**: Given two variables **X** and **Y**, Pearson Correlation Coefficient(PCC) is defined as:

$$\rho_{X,Y} = \frac{cov(\mathbf{X}, \mathbf{Y})}{\sigma_X \sigma_Y} \quad (8)$$

   where $cov(\mathbf{X}, \mathbf{Y})$ is the covariance, σ_X and σ_Y are the standard deviations of **X** and **Y** respectively. $\rho_{X,Y}$ takes the values between -1 and 1.

4 Results and Discussion

Throughout our experiments, we use **50** dimensional word vectors, **50** hidden layer units, learning rate as **0.05**, number of epochs as **21** and a batch size of **1**. The performance of various models on both the datasets is summarized in Table 1. Among the three bi-directional models, LSTM performs the best.

4.1 Correlation Analysis

We analyze the correlation between the hidden layer representations and the weight parameters connecting hidden and output layers. Meeting our expectation, this correlation of hidden layer values is found to be higher with the weight parameters corresponding to the true tag for a given input word. For instance, take a sentence from ConLL dataset: "The students, who had staged an 11-hour protest at the junction in northern Rangoon, were taken away in three vehicles.". Here, the word "Rangoon" has it's true tag as *I-LOC* and rest all

Dataset	Instances			Test Set Performance			
	Training	Validation	Testing	Model	Precision	Recall	F Score
CoNLL-2003	14987	3466	3684	RNN	83.42	81.77	82.59
				LSTM	85.87	84.41	**85.13**
				GRU	85.11	83.66	84.38
JNLPBA-2004	18046	500	3856	RNN	67.71	68.99	68.34
				LSTM	67.94	72.69	**70.23**
				GRU	67.55	70.05	68.78

Table 1: Statistics and performance of different models on two NER datasets used in this work.

are context words. Figure 2 plots the normalized values for left side part of the hidden representation for "Rangoon", along with corresponding weight parameters for *I-LOC* and *I-MISC* tags. *I-MISC* has been chosen as it's corresponding dot product is maximum among all the false tags. The high correlation between the hidden representation and weight parameters for the true tag can be clearly observed from the figure.

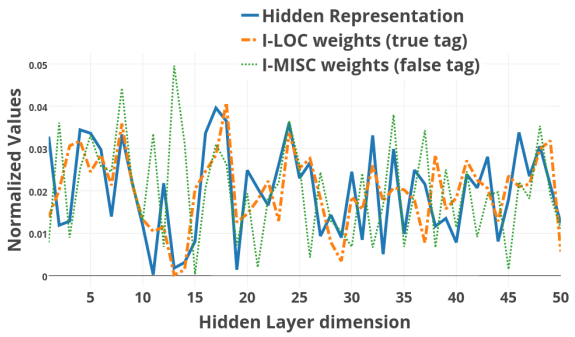


Figure 2: Visualization of hidden representation of a LOC entity word "Rangoon" and weight parameters corresponding to true and false tags.

Table 2 gives the correlation values for above three measures corresponding to the "Rangoon" instance.

Tag	Dot Product	KL Divergence	PCC
I-LOC (True tag)	**7.27**	**0.15**	**0.62**
I-MISC (False Tag)	1.76	0.48	0.17

Table 2: Correlation values obtained corresponding to "Rangoon" instance from CoNLL dataset.

4.2 Analysis of Relevance Scores

In order to evaluate the ability of RNN models to capture important contextual words, we do a qualitative analysis at both word and sentence levels. This section provides instances from both CoNLL and JNLPBA datasets to illustrate how the three measures can be used to identify salient words with respect to bi-directional model. Although we compute word rankings using the three measures described above, our demonstrations in the paper primarily focus on the M_LRC method. M_LRC is able to treat each word individually with due attention to dependency on another words in a given sentence.

At the word level, we further breakdown the visualizations into three types:

Fixing a word and a method: In this case, we fix a particular word and use M_LRC method. We analyze how the importance scores change with various models, entities and correlation measures. Figures 3a, 3b and 3c show heatmaps by fixing the word "midfielder" and M_LRC method for CoNLL dataset. Based on our intuition, the word "midfielder" should have higher importance scores for PER entity. This is clearly visible in the illustrations. All the three correlation measures are able to capture this intuition to a reasonable extent. Similarly, figures 3d, 3e and 3f show heatmaps for "apoptosis" on JNLPBA dataset. The higher scores given to class CT (*cell_type*) are in agreement with the results of M_WF method as well as with our intuition as "apoptosis" indicates cell death. It can also be observed that all the bidirectional models do quite well in both these cases.

Fixing a model and a method: In this case, we fix a particular model and try to visualize how the models score different contextual words for different entity types. Figure 4 shows the heatmaps by fixing RNN, LSTM and GRU respectively with M_LRC method (using dot product). Our intuition that "captain", "city" and "agency" would be relevant for PER, LOC and ORG entities respectively, is proved to be true as can be observed in all of the cases. However, neural models are unable to associate "agency" with ORG as distinctively as in case of "captain" and "city". This can be attributed to frequent occurrence of the word "agency" in the context of words belonging to PER or LOC entities, thereby, confusing the

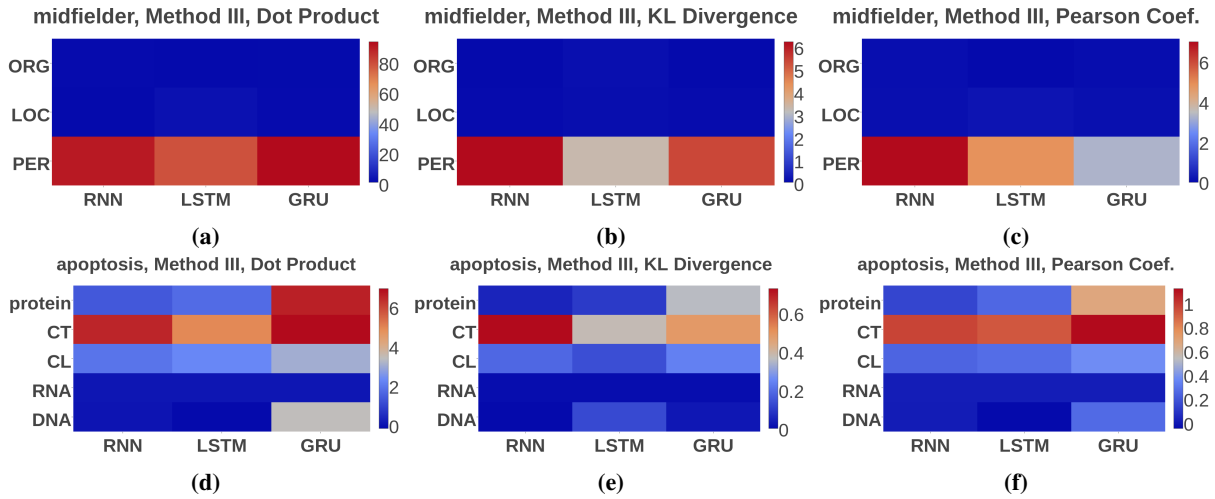


Figure 3: Heatmaps showing the scores for different words across models, entities and methods on CoNLL dataset in part (a), (b) and (c) and on JNLPBA dataset in (d), (e) and (f). Here, CT refers to *cell_type* and CL refers to *cell_line*.

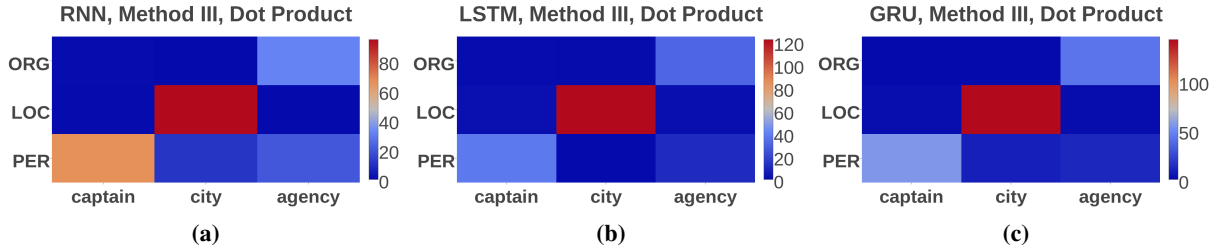


Figure 4: Heatmaps showing the word scores fixing a model with M_LRC method using dot product on CoNLL dataset.

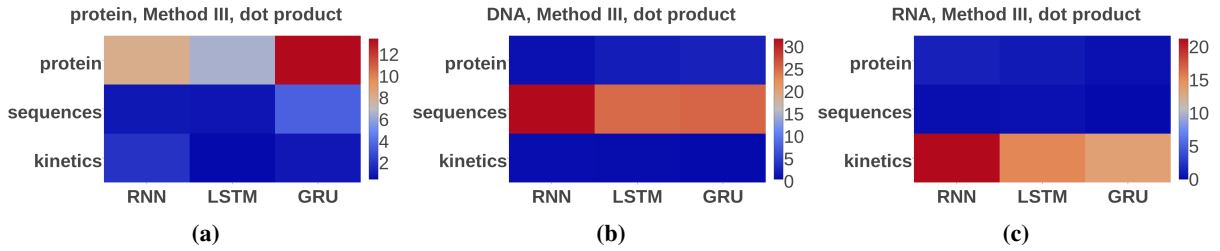
models.

Fixing an entity and a method: Now, we fix a particular entity to analyze which model gives higher importance to different contextual words for a particular entity. Figure 5 shows the heatmaps by fixing entities $protein$, DNA and RNA respectively with M_LRC method. "protein", "sequences" and "kinetics" have high frequency scores for $protein$, DNA and RNA respectively. The models capture this beautifully in all the cases.

At a sentence level, we only consider our best performing model, LSTM. Table 3 gives entity wise word relevance scores for two individual sentences. It uses a sentence from CoNLL dataset - "Saturday 's national congress of the ruling Czech (*I-ORG*) Civic (*I-ORG*) Democratic (*I-ORG*) Party (*I-ORG*) ODS (*I-ORG*)) will discuss making the party more efficient and transparent , Foreign Minister and ODS (*I-ORG*) vice-chairman Josef (*I-PER*) Zieleniec (*I-PER*), said on Friday .". The tags for all entity words are mentioned alongside each word. Notice the high scores for "vice-chairman", "ruling", "congress", "minister" meets the intuitive understanding of these words. Interestingly, round brackets get the maximum scores for M_SLL method, which may be attributed to their frequent use with ORG entity words. Similarly, sentence taken from JNLPBA dataset is: "the number of glucocorticoid (*B-protein*) receptor (*I-protein*) sites in lymphocytes (*B-cell˙type*) and plasma cortisol concentrations were measured in dgdg patients who had recovered from major depressive disorder and dgdg healthy control subjects .". Again, higher scores for "sites" and "plasma" for *cell_type* are in agreement with overall scores given to them.

4.3 Positional effects of context words

In this section, we analyze how the position of context words affects their scores obtained by M_LRC method. We do this analysis for real sentences present in the test sets as well as on artificial sentences. We achieve this by applying the proposed techniques at an individual sentence level. For instance, Table 4 shows the relevant scores of the word "minister" for entity PER obtained by three models, in three test sentences taken from CoNLL dataset. M_WF method indicates that "minister" has high importance for en-



(a) (b) (c)

Figure 5: Heatmaps showing the word scores fixing M_LRC method and entities on JNLPBA dataset.

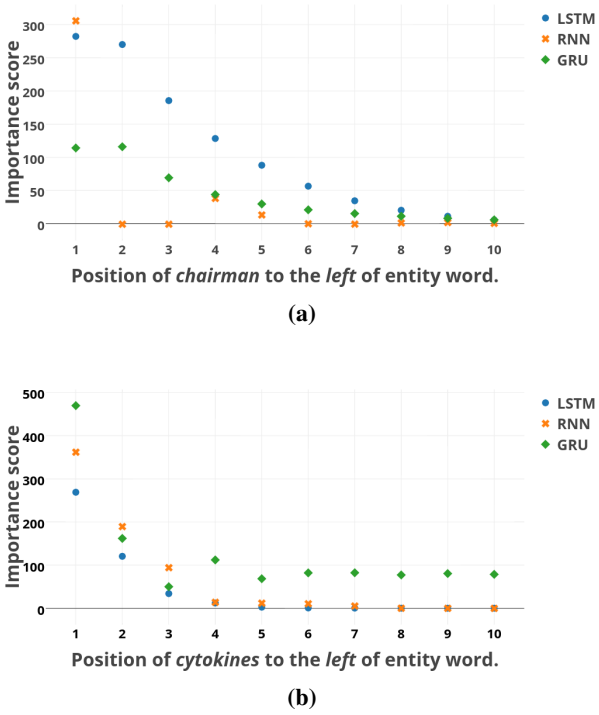
Word	Score
(	9.407
,	8.428
ruling	2.537
vice-chairman	1.41
of	1.203
national	0.901
discuss	0.732
congress	0.728
the	0.723
's	0.486
minister	0.403
and	0.209
saturday	0.065
0	0.03
on	0
friday	0
)	-0.002
said	-0.023
will	-0.045
party	-0.068
making	-0.072
transparent	-0.088
efficient	-0.09
foreign	-0.184
more	-0.202

(a)

Word	Score(Pr)	Score (CT)
control	0	0
and	-0.193	0
major	-0.487	-0.101
number	10.148	2.698
in	0.515	80.745
depressive	7.463	0.039
from	10.221	0.032
had	2.051	0.007
sites	-0.025	18.487
0	0	0
subjects	0	0
plasma	-0.083	0.001
recovered	-0.388	-0.014
cortisol	0.134	0
who	0.933	-0.002
measured	0.639	0.001
healthy	-0.047	0
of	36.08	4.335
dgdg	-0.343	-0.001
patients	3.377	0.007
were	0.454	0.001
concentrations	0.014	0
the	-0.613	2.572
disorder	10.723	0

(b)

Table 3: Entity wise relevance scores for words in two individual sentences using LSTM model: **(a)** Using M_SLL method for CoNLL instance and **(b)** Using M_LRC method with dot product for JNLPBA instance.



(a)

(b)

Figure 6: Position vs relevance score plot for three models for (a) "chairman" w.r.t. PER entity word "Josef" and (b) "cytokines" w.r.t. $protein$ entity word "erythropoietin".

tity type PER matching with our intuition. However "minister" is likely to appear in different sentences with different context and may not have equal relevance as also indicated in the Table 4. In the first sentence, there is no entity word for PER, hence, the score for "minister", corresponding to entity PER is zero. In the second sentence, the score is higher, though not too high as the word is relatively far from the relevant entity word. However, the score is much higher in the third sentence where "minister" is right before the entity words "Margaret Thatcher". Relative scores obtained by using different neural models also match with the general notion that RNN tends to forget long range context (second sentence) compared to LSTM and GRU, and is quite good for short distance context (third sentence).

We further validate the above observation on artificial examples. Figure 6a gives the position verses score plot for the word "chairman" with respect to the PER entity word "Josef". The position tells that how far to the left "chairman" is from the entity word. We create sentences as follows - "chairman Josef .", "chairman R Josef .", "chairman R R Josef ." and so on. Here, R represents a random word. It can be observed that how LSTM and GRU assign a higher score to far off words compared to RNN, justifying their ability to include such words in making the final decision.

Figure 6b shows a similar plot for the word "cytokines" and a $protein$ entity word "erythropoietin" using the same way of creating artificial sentences. Interestingly, GRU assigns higher relevance scores than LSTM and RNN, which is in accordance with the high overall score it gives to

RNN	LSTM	GRU	Sentence
0.0	0.0	0.0	Senegal proposes foreign **minister** for U.N. post .
0.163	2.576	1.031	He was senior private secretary to the employment and industrial relations **minister** from 1983 to 1984 and was Economic advisor to the treasurer Paul Keating in 1983 .
239.793	112.405	199.985	The ODS , a party in which Klaus often tries to emulate the style of former British Prime **Minister** Margaret Thatcher , has been in control of Czech politics since winning general elections in 1992

Table 4: Relevance scores for the word "minister" in three different test sentences from CoNLL dataset.

"cytokines" compared to the other two models.

Rank	Word	Score
1	by	66.162
2	the	22.223
3	in	3.576
4	expression	0.257
5	can	0.222
6	gene	0.221
7	which	0.079
8	over	0.079
9	important	0.003
10	may	0.002
11	establishing	0
12	type	0
13	cell	0
14	0	0
15	specificity	0
16	and	0
17	widening	-0.001
18	range	-0.016
19	recognized	-0.364
20	be	-0.475
21	modulated	-0.534
22	degeneracy	-0.857
23	sequences	-0.917

Table 5: Relevance scores for an individual test sentence from JNLPBA dataset, using LSTM and M_LRC method with dot product.

4.4 Error Analysis

The proposed methods can be effectively used to conduct error analysis on bi-directional recurrent neural network models. For a given sentence, a negative score for a particular word means that the model is able to make a better decision when the word is removed from the sentence. Relevance scores can be used to find out which words confuse the model. Knowing what those words are, is crucial to understanding why the model makes a mistake in a particular instance. For example, Table 5 shows the word importances for the sentence - "the degeneracy in sequences recognized by the otfs (*B-Protein*) may be important in widening the range over which gene expression can be modulated and in establishing cell type specificity ." The LSTM model makes a mistake here by tagging "otfs" with tag *B-DNA*. Words "degeneracy", "sequences", "widening", "recognized" and "modulated" all have a higher overall score for DNA entity class than for $protein$. Hence, the presence of these words in the sentence fool the model into making a wrong decision.

In general, we observe that the presence of words which have high scores for false entity types tend to confuse the model. Position of words also plays a vital role. Words which appear in a far off or a different position than what they generally appear in the training dataset, tend to receive negative or low scores even if they are important. For instance, "minister" mostly appears to the left of an entity word in the training dataset. If, in a test case, it appears to the right, it ends up receiving a low score.

5 Related Work

Various attempts have been made to understand neural models in the context of natural language processing. Research in this direction can be traced back to Elman (1989) which gains insight into connectionist models. This work uses principal component analysis (PCA) to visualize the hidden unit vectors in lower dimensions. Recurrent neural networks have been addressed in recent works such as Karpathy et al. (2015). Instead of a sequence tagging task, they use character level language models as a testbed to study long range dependencies in LSTM networks.

Li et al. (2015) build methods to visualize recurrent neural networks in two settings: sentiment prediction in sentences using models trained on Stanford Sentiment Treebank and sequence-to-sequence models by training an autoencoder on a subset of WMT'14 corpus. In order to quantify a word's salience, they approximate the output score as a linear combination of input features and then make use of first order derivatives. Erasure technique helps us to do away with such assumptions

and find word importances in sequence labeling tasks for individual entities.

Similar to present work, Kádár et al. (2016) analyze word saliency by defining an omission score from the deviations in sentence representations caused by removing words from the sentence. This work, however, targets a different, multi-task GRU framework, learning visual representations of images and a language model simultaneously.

Another closely related work is Li et al. (2016). They use erasure technique to understand the saliency of input dimensions in several sequence labeling and word ontological classification tasks. Same technique is used to find out salient words in sentiment prediction setting. Our work focusing on sequence labeling task has several differences with Li et al. (2016). Firstly, in case of sequence labeling, Li et al. (2016) only focus on feed forward neural networks while our work trains three different recurrent neural networks on general and domain specific datasets. Secondly, their analysis in sequence labeling task is only limited to important input dimensions. Instead, our work focuses on finding salient words which are basic units for most NLP tasks. Lastly, our M_SLL method is an adaptation of their method to find salient words in sentiment prediction task. Unfortunately, for a sequence labeling task, this method is not very suitable. Since it only considers sentence level log likelihood, it makes no distinction between various possible entities such as person or organization. Our M_LRC method, which takes individual word level effects into account, is more suitable.

A significant amount of work has been done in Computer Vision to interpret and visualize neural network models (Simonyan et al., 2013; Mahendran and Vedaldi, 2015; Nguyen et al., 2015; Szegedy et al., 2013; Girshick et al., 2014; Zeiler and Fergus, 2014; Erhan et al., 2009). Attention can also be useful in explaining neural models (Bahdanau et al., 2014; Luong et al., 2015; Sukhbaatar et al., 2015; Rush et al., 2015; Xu and Saenko, 2016).

6 Conclusions and Future Work

In this paper, we propose techniques using word erasure to investigate bi-directional recurrent neural networks for their ability to capture relevant context words. We do a comprehensive analysis of these methods across various bi-directional models on sequence tagging task in generic and biomedical domain. We show how the proposed techniques can be used to understand various aspects of neural networks at a word and sentence level. These methods also allow us to study positional effects of context words and visualize how models like LSTM and GRU are able to incorporate far off words into decision making. They also act as a tool for error analysis in general by detecting words which confuse the model. This work paves the way for further analysis into bi-directional recurrent neural networks, in turn helping to come up with better models in the future. We plan to take our analysis further by including other aspects like character and word level embedding into account.

References

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

[Chung et al.2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[Collobert et al.2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

[Elman1989] Jeffrey L Elman. 1989. Representation and structure in connectionist models. Technical report, DTIC Document.

[Erhan et al.2009] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341:3.

[Girshick et al.2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

[Graves2013] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

[Hong2005] Gumwon Hong. 2005. Relation extraction using support vector machine. In *International Conference on Natural Language Processing*, pages 366–377. Springer.

[Huang et al.2015] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

[Kádár et al.2016] Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2016. Representation of linguistic form and function in recurrent neural networks. *arXiv preprint arXiv:1602.08952*.

[Karpathy et al.2015] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

[Kim et al.2003] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. Genia corpusa semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182.

[Kim et al.2004] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Association for Computational Linguistics.

[Lafferty et al.2001] John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

[Li et al.2015] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.

[Li et al.2016] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

[Luong et al.2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[Mahendran and Vedaldi2015] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5188–5196.

[Mikolov et al.2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al.2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Nguyen et al.2015] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[Rush et al.2015] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

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

[Sukhbaatar et al.2015] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

[Szegedy et al.2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

[Tjong Kim Sang and De Meulder2003] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

[Xu and Saenko2016] Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer.

[Zeiler and Fergus2014] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.