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

Kushal Chawla, Sunil Kumar Sahu and Ashish Anand
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Assam, India - 781039
kushalchawla18@gmail.com, {sunil.sahu,anand.ashish}@iitg.ernet.in

Abstract
Learning algorithms for natural language processing (NLP) tasks traditionally rely on manually defined appropriate contextual features. On the other hand, neural network models learn these features automatically and have been successfully applied for several NLP tasks. Such models only consider vector representation of words and thus do not require efforts for manual feature engineering. This makes neural models a natural choice to be used across several domains. But this flexibility comes at the cost of interpretability. The motivation of this work is to enhance understanding of neural models towards their ability to capture contextual features. In particular, we analyze the performance of bi-directional recurrent neural models for sequence tagging task by defining several measures based on word erasure technique and investigate their ability to capture relevant features. We perform a comprehensive analysis of these measures on general as well as biomedical domain datasets.

1 Introduction
Learning approaches for NLP tasks can be broadly put into two categories based on the way features are obtained or defined. Traditional way is to design features according to a specific problem setting and then use appropriate learning approach. Examples of such approaches include classification algorithms like SVM (Hong, 2005), CRF (Lafferty et al., 2001) among others for several nlp tasks. A major proportion of overall effort is spent on feature engineering itself. The desire to obtain better performance on a particular problem makes the researchers to come up with a very domain and task specific set of features. The major 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 are difficult.

On the other side, neural network models provide a more generalized way of approaching problems in NLP domain. The models try to learn appropriate features themselves. Minimal effort is required in feature engineering. This means that minimal amount of domain knowledge is required. Thus, the techniques developed can be used effectively in various independent problem scenarios. Neural networks have gained huge recognition in recent years and find applications in a variety of areas today (Liu et al., 2017).

Unlike traditional feature based methods, a major drawback of neural network models is that they are more complex. Since the features are not manually defined, it becomes a lot more challenging to interpret them. Neural networks have been applied significantly to various tasks without many insights on what the underlying structural properties are and how the model learn to classify the inputs correctly. Mostly inspired from 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. Analysis is focused at understanding how well the relevant contextual words are being captured by different neural models. The analysis provides a general tool to compare between different models, show 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

We propose three methods to calculate relevance score of context words indicating their importance. 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 \( M \) for a particular task \( T \) in a given dataset.

2.1 Based on word frequency

For a given sentence \( S \in T \) be \( T \). For a context word \( w_c \) in \( S \), let \( L_1(w_c, S) \) be the negative log likelihood of \( T \) 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 entities present in \( S \).

We use the erasure technique similar to (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 \( T \). Let this value be \( L_2(w_c, S) \). Intuitively, if \( S \subseteq S_{w_c,e} \) 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.

\[
I(w_c, e) = \frac{1}{F(w_c, e)} \sum_{S \subseteq S_{w_c,e}} \frac{L_2(w_c, S) - L_1(w_c, S)}{L_1(w_c, S)}
\]

(3)

We refer to this method as Method II in rest of the paper.

2.2 Using sentence level log likelihood

Here we use the sentence level log likelihood to calculate relevance score for each word corresponding to each entity type.

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 \( w \in S \) 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 \( T \). For a context word \( w_c \in S \), let \( L_1(w_c, S) \) be the negative log likelihood of \( T \) 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 entities present in \( S \).

Hence the relevance score is calculated as follows:

\[
I(w_c, e) = \frac{\sum_{S \subseteq D} A(w_c, e, S)}{\sum_{w_c} \sum_{S \subseteq D} A(w_c, e, S)}
\]

(1)

Using inverse frequency to account for irrelevant too frequent words for all entity types, the score can be calculated as follows-

\[
I(w_c, e) = \frac{\sum_{c \in E} \sum_{S \subseteq D} A(w_c, e, S)}{\sum_{c \in E} \sum_{S \subseteq D} A(w_c, e, S)} + k
\]

(2)

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 Method I in rest of the paper.
2.3 Considering left and right word contexts separately

In this case, we work at a word level. 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. With the intuition that the important word 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 $w_e$.

The relevance score $I(w_e, e)$ is then computed by taking average of $L(w_e, w_e, S)$ over all instances. Note the values in above formulae are adjusted first to make all of them positive (shifted dot products) so that same formula works for all cases. We refer to this method as Method III for rest of the paper.

3 Experiments

We considered the task of sequence tagging problem for evaluation and analysis of the proposed methods to interpret neural network models. In particular we chose three variants of recurrent neural network (RNN) models for Named Entity Recognition (NER) task. In NER task, given a sentence, every word in that sentence needs to be tagged with the most appropriate one from a predefined set of all tags.

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.](image)
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. These two datasets, discussed below, are quite diverse as 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, b). 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. We analyze various possibilities to obtain similarities between distributions of hidden layer outputs to the weight parameters. In this study, we consider two other such measures apart from dot product-

- Kullback-Leibler Divergence
- Pearson Correlation Coefficient

4 Results and Discussion

The performance of various models on both the datasets is summarized in Table 1. LSTM model obtained the best performance among the three models. 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. We keep these hyper-parameters fixed for all the experiments for both the datasets.

4.1 Correlation Analysis

First, we analyze correlation between hidden layer outputs and the associated weight parameters corresponding to different tags. As discussed earlier, this correlation should be higher with the weight parameters corresponding to true tag. A sentence - “The students, who had staged an 11-hour protest at the junction in northern Rangoon, were taken away in three vehicles.” is taken from CoNLL dataset for this analysis. “Rangoon” has it’s true label as I-LOC and rest all 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.

Table 2 gives the correlation values for above three measures corresponding to the “Rangoon” instance.
4.2 Analysis of Relevance Scores

We do three kinds of analysis at word level on the two datasets to evaluate the three neural models ability to capture important contextual words indicated by corresponding relevance scores -

Fixing a word and a method: In this case, we fix a particular word and use Method III. We analyze how the word score changes with various models, entities and measures. Figures 3a, 3b and 3c show heatmaps by fixing the word “midfielder” and Method III for CoNLL dataset. According to the output of frequency based Method I, and intuitively, “midfielder” should have higher scores for person entity. This is clearly visible in the heatmaps. 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 Method I as well as with our intuition as “apoptosis” indicates cell death.

Fixing a model and a method: Similarly 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 Method III (using dot product). According to the scores given by Method I, “captain”, “city” and “agency” seem more relevant for PER, LOC and ORG respectively, which can be observed distinctively in almost all of the cases. All the three neural models find it little difficult to associate “agency” with ORG.

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 Method III. “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 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

| Dataset       | Training | Validation | Testing |
|---------------|----------|------------|---------|
| CoNLL-2003    | 14987    | 3466       | 3684    |
| JNLPBA-2004   | 18046    | 500        | 3856    |

| Model      | Precision | Recall | F Score |
|------------|-----------|--------|---------|
| RNN        | 83.42     | 81.77  | 82.59   |
| LSTM       | 85.87     | 84.41  | 85.13   |
| GRU        | 85.11     | 83.66  | 84.38   |

| Tag                  | Dot Product | KL Divergence | Pearson Coef. |
|----------------------|-------------|----------------|---------------|
| I-LOC (True tag)     | 7.27        | 0.13           | 0.62          |
| TMISC (False tag)    | 1.76        | 0.48           | 0.17          |

Table 1: Size and performance of different models on two dataset used in this work

| Tag                  | Dot Product | KL Divergence | Pearson Coef. |
|----------------------|-------------|----------------|---------------|
| I-LOC (True tag)     | 7.27        | 0.13           | 0.62          |
| TMISC (False tag)    | 1.76        | 0.48           | 0.17          |

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

| CoNLL Instance | JNLPBA Instance |
|----------------|-----------------|
| Word           | Score (CT)      | Score (Pr)  | Score (CT) |
| ( )            | 9.407           | 0           | 0          |
| ruling         | 2.537           | major       | -0.487     | -0.101    |
| vice-chairman  | 1.41            | number      | 10.148     | 2.698     |
| of             | 1.203           | in          | 0.515      | 80.745    |
| national       | 0.901           | depressive  | 7.463      | 0.039     |
| discuss        | 0.732           | from        | 10.221     | 0.032     |
| congress       | 0.728           | had         | 2.051      | 0.007     |
| the            | 0.733           | sites       | -0.025     | 18.487    |
| s              | 0.486           | 0           | 0          | 0         |
| minister       | 0.403           | subjects    | 0          | 0         |
| and            | 0.209           | plasma      | -0.083     | 0.001     |
| saturday       | 0.065           | recovered   | -0.388     | -0.014    |
| 0              | 0.03            | cortisol    | 0.134      | 0         |
| on             | 0.933           | who         | 0.933      | -0.002    |
| friday         | 0.619           | measured    | 0.619      | 0.001     |
| )              | -0.002          | healthy     | -0.047     | 0         |
| said           | 0.023           | of          | 36.08      | 4.335     |
| will           | -0.045          | dgdg        | -0.343     | -0.001    |
| party          | -0.068          | patients    | 3.377      | 0.007     |
| making         | -0.072          | were        | 0.454      | 0.001     |
| transparent    | -0.088          | concentrations | 0.014     | 0         |
| efficient      | -0.019          | the         | -0.613     | 2.572     |
| foreign        | -0.184          | disorder    | 10.723     | 0         |
| more           | -0.202          | -           | -          | -         |

Table 3: Entity wise relevance scores for words in two individual sentences using LSTM model using Method II and Method III with dot product for CoNLL and JNLPBA instance respectively.
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 Method III using dot product on CoNLL dataset.

Figure 5: Heatmaps showing the word scores fixing Method III and entities on JNLPBA dataset.
Table 4: Relevance scores for the word "minister" in three different test sentences from CoNLL dataset.

| 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. |

Josef (I-PER) Zieleniec (I-PER), said on Friday.”. The tags for all entity words is 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 Method II, 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. Refer to Appendix for several other visualizations.

4.3 Positional effects of context words

In this section, we analyze how the position of context words affects their scores obtained by the Method III. 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. Method I indicates that “minister” has high importance for entity 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 repre-
sents 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 “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 Method III 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 otf’s (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. We show a few more instances in the Appendix.

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 NLP. (Elman, 1989) visualize embedding by projection to a lower dimensional space. (Karpathy et al., 2015) use character level language models for analysis of recurrent neural networks, restricting the study to a few special cases. Some efforts involve human users choosing odd words out of a given list to interpret semantic dimensions (Murphy et al., 2012; Fyshe et al., 2015). (Kádár et al., 2016) use a linguistic viewpoint to analyze the activation patterns of recurrent neural networks.

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).

(Li et al., 2015) approximate the output score as a linear combination of input features and find saliency scores of features. 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) but to the best of our knowledge, there is no research in attention based models for sequence tagging problems. Our work is closely related to (Li et al., 2016). They also make use of erasure techniques to analyze neural networks but for sequence tagging, their extent is limited to only window based feed forward networks.
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

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

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.

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.

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

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

Alona Fyshe, Leila Wehbe, Partha Pratim Talukdar, Brian Murphy, and Tom M Mitchell. 2015. A compositional and interpretable semantic space. In HLT-NAACL. pages 32–41.

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.

Alex Graves. 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

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

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

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

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.

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

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

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

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

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

Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. 2017. A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26.

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

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

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.

Brian Murphy, Partha Pratim Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. Association for Computational Linguistics.

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.

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. [http://www.aclweb.org/anthology/D14-1162](http://www.aclweb.org/anthology/D14-1162).

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

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

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

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*.

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*. Edmonton, Canada, pages 142–147.

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*. Springer, pages 451–466.

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