IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 17, 2021, accepted June 28, 2021, date of publication July 7, 2021, date of current version July 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095412

Deep Learning Approach for Negation Handling in Sentiment Analysis

PRAKASH KUMAR SINGH AND SANCHITA PAUL
Birla Institute of Technology, Mesra, Ranchi, Jharkhand 835215, India
Corresponding author: Prakash Kumar Singh (singhpraka@gmail.com)

ABSTRACT Negation handling is an important sub-task in Sentiment Analysis. Negation plays a significant role in written text. Negation terms in sentence often changes the polarity of entire sentence from positive to negative or vice versa, resulting in the opposite meaning of the sentence than what is observed by the machine learning based linguistic model. As automatic opinion mining has become very important in this digital era, proper handling of negation term is the need of the hour. In any natural language negations can be formulated both explicitly or implicitly while their use is very much domain-specific. Existing negation handling techniques follow rule-based approach and mainly used in medical domain. Due to the complex syntactic structure of negation, it is hard to build general purpose machine learning based negation handling model on user review or conversational text data. In this paper, we investigate negation components i.e., cue and scope in a sentence which determine the polarity shift in sentence. We propose LSTM based deep neural network model for negation handling task where the model automatically learns the negation features from labeled input training dataset. We used *ConanDoyle* story corpus for model training and testing, which is pre-annotated with negation information. The proposed model first identify negation cues in each sentence and then using bidirectional LSTM extracts the relationship between cue and other words to identify scope of the cue in sentences. We derived word level features for model training to determine correct polarity of the sentence. Result shows that the LSTM based nonlinear language models perform comparatively better than the traditional state of the art SVM, HMM or CRF based models. BiLSTM achieved best result, F1 measures 93.34%, outperform traditional rule based model in negation handling task.

INDEX TERMS Negation cue, scope, sentiment analysis, feature embedding, recurrent neural network, LSTM, attention learning.

I. INTRODUCTION

Negation is a linguistic phenomena in natural language which reverse the meaning of the sentences. It often reverses an affirmative sentence into negative negation which affect the polarity of a word, hence sentiment expressed in text also changes accordingly. Negation handling is an important sub-task in sentiment analysis in natural language processing (NLP) and considered as one of the hardest problem in NLP in opinion mining. Negation handling in NLP deals with automated detection of polarity shift in opinion expressed in natural language text format. To deal with various NLP problems, there are many statistical, probabilistic and rule based approaches providing automated solution to the problems.

Early approaches perform negation handling tasks mainly uses rule based approach. The rule-based approach effectively performed negation handling in medical domains and rarely need fine-tuning. This approach is mainly suitable in a situation when we already have set of predefined negations. Since medical report text data is not much affected by the linguistic complexity, rule based approach work best in this domain. NegEx, ConTex, pyContext, cTAKES and DEEPEN are few examples of rule based negation handling algorithms used in medical domain. In NLP terms, these negation handling approaches can further be divided into lexicon based and syntax based, where the first three are the lexicon based and the other two are syntax based. Much of the early work on the negation handling has been focused on the medical domain. Plus point with medical domain is that clinical data follow standard format in it's reports which usually do not have natural language complexities. Thus, negations present

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Hao Chen.

in the clinical reports can be seen as static in nature, therefore can easily be modeled by defining rules. On the other side linguistic complexities can easily be seen frequently in general purpose user review data, so automated negation handling on the review data is relatively far more difficult to implement.

Importance of negation handling task in real world can be seen in sentiment analysis. Growing use digital platforms like social media, blogs etc., has left Internet with huge volume of unstructured text data containing user opinion. In today's digital era, every organization has their digital presence where they reach out to their users for providing better services. This digital revolution has forced organizations to use NLP applications like chat-bot or opinion mining to be relevant in today's competition. The problem of negations can easily be seen in such application that can be resolved by mean of simple rule based approach resulting in urgent need of machine learning based negation handling technique to be developed. Scarcity of negation-annotated natural language corpora has remain a bottleneck for NLP research community in developing machine learning based model in negation handling tasks in sentiment analysis. Seeing the challenge researchers started annotating text data with negation information for machine learning community and released it for research. Recent advancements in the uses of deep learning based solution to various language modeling tasks has drawn attention towards the development of neural network based model for negation handling problem.

There are many NLP based approaches that help analyzing users review data automatically by means of applying machine learning algorithms. Analyzing opinion data using machine learning and NLP techniques involve many high and low level sub-tasks in sentiment analysis. Sentiment analysis task at the document level is often seen as classification problem with the aim of classifying the document into positive or negative class. Classification of the documents are based on the polarity associated with individual word in sentences. Based on the overall document's polarity, review document is said to have positive or negative sentiment. It is observed that this approach deals with the Bag of Words features in NLP where model does not pay special attention on the negation in context with other words in sentence. Some frequently used negations are "no", "not", "never" etc, which occur very frequently in written text. There are two main components of negation in any sentence, "cue" and "scope". Cues are the words used in sentence which changes the meaning of other words or the entire sentence. Scope is related with cue which indicates part of sentence impacted by cue in the sentence i.e., the consecutive block of words in sentence which realizes polarity shift due to cue. For example

[I trust that there is] *nothing* [of consequence which I have overlooked]

In the above example, cue term in the sentence is "nothing" and the scope i.e., the part of sentences affected by the negation cue "nothing" is "of consequence".

Negation handling is a low level sub-task in sentiment analysis which deals with identification of cues and it's scope

TABLE 1. Negation cues lexicon.

Negation Lexicons				
no	not	cannot	cant	aint
dont	didnt	nor	none	hadnt
oughtnt	hasnt	havent	havnt	isnt
neednt	neither	never	nobody	horrible
hardly	lacks	darent	dislike	lacking
doesnt	nothing	nowhere	mightnt	mustnt

at sentence level. The semantic computation of negative sentences are more complex than the affirmative one. Negations at sentence level in user review documents, not only appear in short phrases like, "no longer", "no more", "no way", "by no means" etc., but many a time these negations have long run dependencies, for example negations that appears in "neither … nor" format, which shift the polarity of words in sentence sitting at long distance. Quite often cues appears in contracted form in sentences, like can't, couldn't, isn't, wasn't, aren't etc. Complex structure of negations are often difficult to handle by simple rule based linguistic model. Proper modeling of the cues and its span in sentence has direct impact on the accuracy in sentiment analysis task [1], [2] [3]. Scope detection is even more complicated than identifying cues in sentences. Reference [4] thoroughly studied the distribution of cues and their syntactic structure in WSJ(Wall Street Journal) corpus. Some frequently occurring negation cues are shown in Table 1. Scope of cues varies a lot in sentences, it may be limited to the neighboring word or sometimes it may extend to the end of sentence. For example, in a movie review sentence, "The movie was not interesting" the cue term is "not", while the scope of the cue term "not" is just next word after cue, i.e., the term "interesting". But in other sentences like "I cannot call this film a worth watching movie" the effect of the negation cue "not" is until the end of the sentence. The original meaning of the words changes if a word with positive or negative polarity falls inside the scope of negation. Reference [5] performed detail analysis on the scope of negations in Spanish language and its impact in sentiment analysis on tweets.

There are some negation terms that can be easily identified and modeled by applying simple rule based algorithms, while many others are implied negations and require additional linguistic features associated with negation terms to be learned by the model. Many words occurs in sentences which are not explicitly a negation term but carry negative sentiments. Certain prefixes and suffixes are there which often flips the polarity of the words completely, resulting in higher false positive for the model. These rule based models are completely domain dependent and cannot be generalize to other domains.

A. EXPLICIT AND IMPLICIT NEGATION

Explicit negations are those negation cues or words which have a negative meaning associated with words. These negation cues are also assigned negative sentiment score in

TABLE 2. Affixes as negation cues in reviews.

prefixes	Effect	Morpheme	suffixes	Effect	Morpheme
in	Negation	Prefix	less	Negation	Suffix
accurate	Verb	Root	hope	Verb	Root

TABLE 3. Common prefixes and suffixes as negation cues.

prefixes	root word(+ve)	polarity shift(-ve)	suffixes	root word(+ve)	polarity shift(-ve)
im	perfect	imperfect	ness	dark	darkness
ir	rational	irrational	sion	creative	creativity
il	legal	illegal	able	believe	believable
non	sense	nonsense	er	big	bigger
dis	like	dislike	est	tough	toughest

sentiment lexicon. On the other hand the implicit negations does not necessarily have clear negation polarity but implied meaning of the sentence clearly seems to have negative sentiment. For example in the sentence, ''this movie is not good'', the term ''not'' is an example of explicit negation. In another example sentence, ''with this act, it will be his first and last movie'', even though there is not even a single negative cue present in sentence, yet it has implied negation. Reference [6] discussed in detail about the role and impact of explicit and implicit negations in sentences.

B. MORPHOLOGICAL LEVEL NEGATION

Morphemes are the basic building blocks of the words. This includes prefixes, root word and suffixes. Negation can be present at morphological level in the form of affixes like **un**-*happy* or *sense*-**less**. These prefixes and suffixes often shift the polarity of the root words in sentences and should be treated as negation terms as it has great influence the document classification task. Commonly used prefixes and suffixes are shown in below tables at syntactic level like ''less'', ''un'', ''in'' etc., that contribute in polarity shift. Table 2 shows morphological break down of words inaccurate and hopeless. The two words are example of implied negations. Frequently used suffixes and prefixes in the data set is shown in Table3.

These explicit, implicit or morphological level negation indicates that the scope of negation is not fixed in sentences, they differ depending on distinct linguistic characteristics such as punctuation marks, part of speech (POS) tags, conjunctions etc.

Negation handling on raw data require many steps like preprocessing, feature representation, model testing and evaluation etc. At each step it require deep understanding of syntax and semantic understanding before making any selection from the choices available to perform a specific task. Be it tokenizer, embedding, activation function or weight initializer parameters, it must be chosen carefully in an end to end negation handling model. A General Architecture for end to end Negation Handling task is shown in below Figure1.

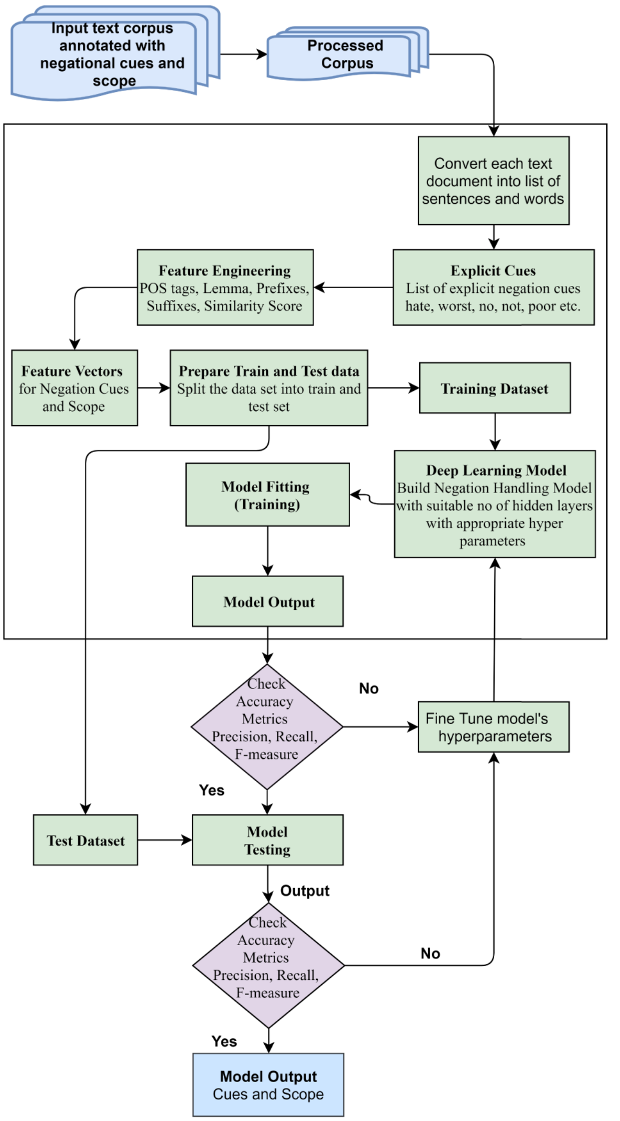


FIGURE 1. General Architecture for Sentence Level Negation Handling.

II. EXISTING APPROACH FOR NEGATION HANDLING

Negation handling techniques in real world use case mainly followed rule based approach. Due to linguistic complexity rule based approach seems feasible to implement within its limited scope. However some efforts has also been made using statistical machine learning based approach. Negation handling was seen as a problem in the medical domain first, resulting in early efforts made in this direction on the clinical data set of patients' discharge summary [7], [8] [9]. These rule based approaches are implemented using regular expressions which require significant human efforts in creating features and analyzing rules to cover maximum possible negation aspects. To overcome complexity in rule based techniques, machine learning based approach has been adapted on the BioScope medical dataset using IGTREE algorithm by [10], showing significant improvement in model performance as compared to the regular expression models. [11] performed scope detection task on movie review dataset using SentiWordNet, a sentiment lexicon. Author suggested that in spite of the presence of negation cue and subsequent polarity shift, it cannot be treated as the affirmative counterpart of the word i.e., ''not bad'' and ''good'' cannot be treated equally in terms of polarity. Despite the success of rule based techniques in medical domain, not much application of rule based approach is seen on the general text data. Lack of negation annotated text data resulted in slow progress in machine learning side in negation handling direction on general text corpus. Also machine learning based models require sufficiently large labeled data in model fitting. Even with the availability of dataset in medical domain, machine learning based model has not been tested to the extent where the model can be generalized. Taking the inspiration from medical domain and following the annotation guidelines of medical dataset Bio-Scope [12], recently some efforts has been made in annotating text data with negation information. One such general text data annotated with negation cues and scope is released by [13] ''SFU Review Corpus Negation Speculation'' by the linguistic department of Simon Fraser University (SFU). Another dataset ''ConanDoyle-neg'' is released by the Linguistic Department CliPS- University of Antwerp [14] which is publicaly available for research in negation handling. Rule based approaches may be a feasible solution in certain situations where negations are static in nature and identifying patterns and deriving rules are not much complex. In reality, negations are inherently dynamic due to its association with linguistic complexity. Having rule based approach provide a solution but within its own constraints. In contrast to the rule based approach, machine learning based approach automatically learns the features associated with negations. Given sufficiently large labeled training data, machine learning model can learn greater negation variations, making the model more efficient and reusable. Negation components cue and scope identification can be formulated as classification and sequence labeling task. Support Vector Machine (SVM) based machine linear model has shown good result in many classification tasks in NLP applications, making it default choice for cue identification problem, whereas for scope detection problem, Hidden Markov Model(HMM) and Conditional Random Field(CRF) are the two algorithms producing best result in sequence labeling tasks. HMM and CRF both are Markov chain-based model often fail to handle long sequential dependency, due to their Markov assumptions, which states that the dependencies of the input sequence longer than three steps are often ignored, while in real case, scope of cue has often long run dependencies in sentences. We proposed a deep neural network based negation handling model to provide solution to this long run dependency issue by using recurrent neural network (RNN) and its variants, which store context information in cell memory to resolve dependency issues. There are many variant of RNN e.g., LSTM, GRU and BiLSTM which are the enhancement of the standard RNN architecture.

III. RELATED WORK

Early efforts on negation handling in text document (medical report) carried out by [7] to find negation along with the disease mentioned in discharge summary of patient medical report. Author proposed an algorithm NegEx which uses regular expression to identify the scope of negation terms such as ''no'', ''ruled out'' in discharge report to find if the disease mentioned is negated or not. NegEx achieved 94.5% specificity as compared to the then existing base line approach having 84.5% accuracy. Reference [7] proposed LALR(1) context free grammar based negation handling algorithm Negfinder, to recognizing negated pattern in the clinical dataset. [15] identified the scope issue and discovered that negation cues does not necessarily always flip the polarity of every next words in sentences. There are some parts of sentence under the influence of the negation cues. Extending the NegEx approach [16] proposed another regular expression based algorithm ConText to identify scope of the negation term in discharge summary to find the historical disease connection with the present disease. Author suggested three contextual values in addition to NegEx's negation, that are hypothetical, historical, and experiencer. Working in same line on Medical Text mining problem [12] released BioScope medical data corpus that serve as a baseline for annotating corpus with negation cue and scope for negation handling task in other domain also after it appeared in CoNLL2010 shared task. [17] used dependency parser for negation detection in clinical dataset to improve the performance of cTAKES (Clinical Text Analysis and Knowledge Extraction System) which proved to be much better in handling linguistic complexity. Reference [9] used Stanford dependency parser to established relationship between negation words within sentences to reduce NegEx false positive. Scope detection task using rule-based approach and described many rules to handle different types of negation and proposed terminology ''scope'' of the negation term ''cue'' in the sentence as the part of the sentence affected by negation. Reference [1] investigated the impact of negation term ''not'' using typed dependencies parser and some static and dynamic delimiters and proposed some heuristic rules involving sentimental verbs, sentimental adjectives, sentimental nouns, double object rules for scope detection, resulted into improved accuracy of the sentiment analysis model. Author also discussed about some exceptional cases where negation terms did not have any scope. They computed feature vector of length eight to define scope candidate for negation term and feed these feature vectors to $C4.5$ decision tree, resulting in 88.4% accuracy. For handling negation cue on general text data, [14] released *ConanDoyle-neg dataset* annotated with negation cues and scope. Dataset is based on Conan Doyle famous stories, ''The Hound of the Baskervilles (HB)'' and ''The Adventure of Wisteria Lodge (WL)''. The corpus serves the baseline for identifying impact of the negation terms in text data. [2], [3] discussed in detail about different types of negations present in sentences and its impact in sentiment analysis task. They observed that major challenges faced in sentiment analysis are the presence of valence shifter terms which act as negation cue and has a great influence in determining polarity of the document. [11] explored role of contextual valence shifter in detail while performing sentiment analysis. Implied negations are often difficult to identify while they are present across all the review documents. [18] presented quantitative analysis report on the implied negation based on his analysis on six different corpora and found that implied negations widely present in written text. [19] explored negation terms and its scope in context of the sentiment analysis.

A sentence is syntactically divided into noun phrase and verb phrase which in turn can be broken down further into constituent sub phrases with leaf node as POS tag for each term in the sentence. Relationship among these constituents must be captured in order to address the polarity shift in the sentences. Since the effect of negation terms are primarily observed at sentence level, sentence structure needs to be understood by the algorithms so that the linguistic model should be able process the syntactic level negation cues to interpret correct semantic. There are several parser which extracts the relationship among words in sentences. [20] handled negation cue present at syntactic and morphological level and observed better result in sentiment analysis task. Sentences in a review documents can be classified as subjective or objective sentences [21], [22]. A sentence is called as subjective sentence if it does contain opinion words, while other sentences having facts only is objective sentence that do not play major role in determining polarity of the documents. Ignoring objective sentences not only reduces the feature size significantly but also improve accuracy of the sentiment analysis model. They proposed Graph based technique to find subjective sentences in the review documents.

Reference [23] used machine learning approach in negation handling. They used conditional random field (CRF) based model trained on the features extracted with dependency parser on BioScope and Product Reviews dataset. Reference [24] performed negation scope detection on twitter dataset by exploiting valence shifter in the tweets. Other than probabilistic model, SVM has shown good results in negation handling task. [25] claimed SVM machine learning algorithm performs better than rule based approaches in identifying scope in SFU review corpus dataset.

Automated approaches to negation handling mainly rely on the key observation that a sentence can be regarded as sequences of events, i.e. words, in which a word occurrence depends upon the long sequence of the previously occurred words. Such sequences of observations, could be approached as a sequence labeling task in which each word is labeled as being part (or not) of the negation scope associated with a negation cue word. Reference [26] proposed SVM and CRF based linguistic model to handle negation. they used support vector machine(SVM) for cues classification based on maximum margin, while scope resolution was done by transforming it into sequential labeling problem using Conditional Random Field (CRF). Though maximum approaches in scope resolution problems are solved using Hidden Markov Model (HMM) and (CRF), yet these probabilistic model fails in capturing long run dependency. When dependency gets longer HMM model becomes complex to get these resolved while CRF requires a lot of hand-crafted feature to be generated before it can be used, resulting in low efficiency in handling the scope problem.

A. NEURAL NETWORK BASED APPROACH

Neural network has shown its' significant importance in solving complex linguistic tasks on text data, specially in sequential modeling tasks in NLP e.g., in machine translation [27], and sequence labeling tasks [28]. Reference [29] used recurrent neural network in identifying the scope of the negation cues without explicitly doing any feature engineering in solving the negation problem. There are many variants of standard RNN, like LSTM [30] and GRU [27]. Bi directional LSTM (BiLSTM), is another variant of LSTM, that seems quite promising in sequence modeling tasks [31] by leveraging context information in sentences. [32] presented a recursive neural network sequence labeling model for negation handling task that learns syntactic information automatically global dependency tree. Model learns high level representation of words with context information from sentences and captures all cues and scope successfully. Results on medical dataset BioScope and Chinese dataset CNeSp show that model outperforms the state-of-the-art model. [33] proposed reinforcement learning based approach to detecting, understanding and interpreting negations in natural language. Authors performed document level negation handling and eliminated the need of expensive word-level annotations on financial dataset. They concluded that the model performance matched with human interpretation of negation and obtained significant improvements over rule based techniques. [34] proposed heuristic approach for negation handling via dependency graphs. They proposed an algorithm for negation detection based on grammatical distance from a cue word in a typed dependency graph.

IV. MACHINE LEARNING ALGORITHMS FOR SEQUENTIAL DATA

Scope detection problem can be treated as sequential labeling problem. Probabilistic models like Hidden Markov Model (HMM) and Conditional Random Field (CRF) deal with such sequential data with their own pros and cons in handling the complexity of the task.

A. PROBABILISTIC MODELS

1) HIDDEN MARKOV Model(HMM)

HMM deals with the observed event, in sentence words and the hidden events i.e., the label we want to assign to the word. It is a Generative probabilistic model. Basic assumption of first order Markov Chain is that in sequential data, current state depends on the previous state.

$$P(q_i|q_1, q_2, \ldots .q_{i-1}) = P(q_i|q_{i-1}) \quad (1)$$

While first order Markov model HMMs assumes complete independence where any observation o_i depends only on the state that produced it i.e.

$$P(o_i|q_1..q_i,..q_T,o_1,..,o_i,..o_T) = P(o_i|q_i) \tag{2}$$

Given input sequence, this model is used to predict observation that is most likely. HMM model work on the principle of Forward and Backward algorithm that scans word sequence from left to right and right to left respectively to capture the contextual information for a given word. Though HMM work on the principle of joint probability, other variant of the model is based on Maximum Entropy (MaxEnt) principle, is called Maximum Entropy Markov Model.

2) CONDITIONAL RANDOM FIELD (CRF)

CRF is much powerful than HMM, in fact whatever we can do with HMM is also doable in CRF. Advantages of CRF over HMM is described in detail in [15]. CRF is discriminative model used for sequential data. It first defined feature function, initialize with random variable and then using Gradient Descent update the value to get

$$P(y,X,\lambda) = \frac{1}{Z(X)} exp\{\sum_{i=1}^{n}\sum_{j}\lambda_j f_i(X,i,y_{i-1},y_i)\} \tag{3}$$

where

$$Z(X) = exp\{\sum_{y'\in v}\sum_{i=1}^{n}\sum_{j}\lambda_j f_i(X,i,y'_{i-1},y')\}$$

is maximum likelihood estimate to calculate λ value.

Negative log likelihood of CRF probability distribution is calculated by formula.

$$\begin{aligned}
&L(y,X,\lambda)\\
&= -\log\{\prod_{k=1}^{m}P(Y^k|x^k\lambda)\}\\
&= \sum_{k=1}^{m}\log[\frac{1}{Z(x_m)}\exp\{\sum_{i=1} n\sum_{j}\lambda_j f_i(X^m,i,y^k_{i-1},y^k_i)\}]\\
&\quad \times\frac{\partial y}{\partial x} = -\frac{1}{m}\sum_{k=1}^{m}F_j(y^k,x^k)+\sum_{k=1}^{m}p(y|x^k,\lambda)F_j(y,x^k)
\end{aligned}$$

where $F_j(y,x)$ is the partial derivative with respect to lambda λ

$$F_j(y,x) = \sum_{i=1}^{n} f_i(X,i,y_{i-1},y_i)$$

For updating parameters Gradient Descent is used with small step size until the value converges to the optimum.

$$\lambda = \lambda + \alpha[\sum_{k=1}^{m}F_j(y^k,x^k)+\sum_{k=1}^{m}p(y|x^k,\lambda)F_j(y,x^k)]$$

B. NEURAL NETWORK FOR SEQUENTIAL DATA

In Basic architecture of RNN output depends on the current input and previous output, giving it capability to look back into the past for context. It treats each input sentence as a sequence of words. Each word w_i appearing at timestamp t_i is represented in d dimensional vector input to the RNN's, one word at a time.

From feature representation perspective in text data, significant advancement has been achieved in form of embedding techniques, like words embedding, characters embedding. Sentence and paragraph embedding are some of the latest development in representing document as vector, that has shown significant potential in various NLP tasks. These embedding techniques are used to capture contextual information in the text data which is further used as input to the recurrent neural network (RNN) based linguistic model. RNN is basically useful in handling long run dependency, however, the basic RNN architecture, suffer from vanishing and exploding gradient issue while back propagating error towards the initial layer of the network during back propagation through time (BPTT). Further improvement in basic RNN resulted in LSTM architecture which solved the vanishing gradient issue with the help of gated input, output and update operation. LSTM neural network success in retaining long and short term information in memory enabled it to solve long run dependency issue with the text data, which proved to be quite helpful in negation handling task. Though LSTM seems computationally expensive in terms of number of weight parameters need to be learned during training phase, but the model seems to have produced satisfactory result in sequential task. Extension of LSTM came in form of GRU, which basically combined hidden and cell state and forget and input gate of LSTM resulting in fast convergence of the model, without compromising with accuracy. The evolution in the field of neural network is continue in form of transformer model [35], encoder decoder based attention mechanism, that theoretically seems quite appealing in solving negation handling task, but progress is little slow in this direction. In a naive term, Neural Network based Language Model can be defined as

$$P(w|h) = \frac{e^{\sum_{i=1}^{N}\lambda_i f_i(s,w)}}{\sum_w e^{\sum_{i=1}^{N}\lambda_i f_i(s,w)}} \tag{4}$$

where, s is hidden layer state, f is set of features, λ is set of weights, h is history or context word. Computational complexity of Neural Network Language Modeling is quite high that can be defined as

$$I \times W \times ((N-1)\times D\times H + H\times V) \tag{5}$$

where, I is the number of training epoch before convergence. W are the words/tokens, H is hidden layer size and D is dimensionality of words.

1) FEATURE EMBEDDİNG FOR INPUT TO THE NEURAL NETWORK

Recurrent Neural Network takes input data as sequence of words at different timestamp, where the total timestamp is equal to the size of vocabulary. These input words need to be represented in numeric vector format of fixed dimension containing the contextual information captured from neighborhood. There are two ways in which cues and non-cue words can be represented in an embedded vector format. One Hot Encoding and word embedding. One Hot Encoding is sparse vector representation of words in very high dimension $|V|$ (i.e., vocabulary size). Problem with one hot encoding vector is, it does not constitute contextual information of. Consider two sentences "have a nice day" and "have a great day". Even if the two sentences are quite similar, using one hot encoding approach, they do not find any similarity between them, since all vectors for each individual term are orthogonal and do not share any contextual information. To overcome this issue [36], [37] introduced neural network based word embedding algorithms, CBOW (Continious Bag of Word) and SKIP-GRAM. Purpose of using word embedding is that, it not only represent a word as vector but also contains contextual information, where context is the surrounding words for any given cue in the sentences. Given the size of context window, based on the maximum length of sentence in negation dataset, CBOW word embedding model learns the vectors for all the negation cues. While skip-gram does just opposite of this and learns the context vector for a given input word. Given a sequence of words, skip-gram model tries to maximize the log likelihood.

$$\sum_{t=1}^{T} \sum_{c \in C_r} logP(W_c|w_t)$$

The problem of assigning cue or non-cue label, given the input words, is treated as binary classification problem. If f is function that assigns real valued score to word pair, (w_t, w_j) then probability of the context word is defined using soft max function i.e.

$$P(w_c|w_t) = \frac{e^f(w_t, w_c)}{\sum_{j=1}^{W} e^f(w_t, j)}$$

Word embedding performs well in negation handling task, as it represent each word in dense feature vector, where elements of feature vector represent context information withing a window of specified length.

2) RECURRENT NEURAL NETWORK (RNN)

RNN is neural network based machine learning algorithm, which is recursive in nature and capable of feeding learn patterns from input sequences at time t_{i-1} to t_i accept sequence of input over time, the feature can be utilized in dealing with the text data. Since each sentence is basically sequence of words. Sequence-to-sequence learning can be used in different use cases and mainly has been used in POS tagging where each token in the sentence is assigned a POS tag. As we can see in the below diagram of RNN, it basically has a loop that we can think of in expanded format as shown. During the training RNN learns matrix values that it keep on adjusting using gradient descent technique. It accept input and output in a fixed length vector format so raw input data need to be transformed in such a way that can be fed to the RNN.

$$h_t = \tanh(W_{hh}^t h_{t-1} + W_{xh}^t h_t) \tag{6}$$
$$y = W_{hy}^t h_t \tag{7}$$

where

h_t is the hidden sate learn by RNN at timestamp t.

W_{hh} is the weight matrix from one hidden state to the next the hidden sate that is learn by RNN at timestamp t.

W_{xh} is the weight matrix from input to the hidden state.

W_{hy} is the weight matrix from hidden to the output state.

There are different types of RNN models for handling sequential problems, like many to one, one to many, many to many which suites for different linguistic tasks, and produces solution as per the type of problem. Based on the types of model, it accept input at different timestamp and produces output either at each timestamp or at the end of the timestamps. We used many to one sequential model for cue classification in negation handling. For determining scope of the cue, we used many to many model.

3) LONG SHORT TERM MEMORY (LSTM)

Since vanishing gradient issue is prevalent in RNN in dealing with the long sentences [26] specially in *nither ... nor* type of negation cue we chose variant of it i.e., LSTM based approach to build the negation handling model. In this method, a multilayer encoder mapped a sequence of inputs onto a fixed dimension vector and another multilayer decoder construct target sequence from the learned vector. LSTM is slightly different than RNN in the sense that LSTM network uses gates (forget, input and output) to forget irrelevant information learned from past timestamp, update with the current information and store new updated information in the cell which serve as long term memory which is represented as cell state c_t. Short term memory is stored in hidden state h_t of LSTM network. It uses sigmoid function (σ) that squashes values between 0 and 1 that helps the network in deciding amount of information to forget and retain while building long term context. It also use (tanh) function that generate new information to be added to the cell state. Pointwise multiplication ($\odot$) and addition ($\oplus$) in determining the final cell state i.e., the content of long term memory.

$$forget\ gate : f^{(t)} = \sigma(W_f x^t + U_f h^{(t-1)}) \tag{8}$$
$$input\ gate : i^{(t)} = \sigma(W_i x^t + U_i h^{(t-1)}) \tag{9}$$
$$\tilde{c}^{(t)} = \tanh(W_c x^t + U_c h^{(t-1)}) \tag{10}$$
$$output\ gate : o^{(t)} = \sigma(W_o x^t + U_o h^{(t-1)}) \tag{11}$$
$$hidden\ state : h^{(t)} = o^{(t)} \odot \tanh(c^{(t)}) \tag{12}$$
$$cell\ state : c^{(t)} = f^t \odot c^{(t-1)} + i^t \odot \tilde{c}^{(t)} \tag{13}$$

where

W_i and U_i is the weight matrix associated with the input layer.

W_f and U_f is the weight matrix associated with the forget layer.

W_o and U_o is the weight matrix associated with the output layer. W_c and U_c is the weight matrix associated with the LSTM cell.

4) ATTENTION ON SUBJECTIVE SENTENCES AND CUE WORDS

Usually, each review document is composed of multiple sentences. The review document express users' overall opinion but it does not necessarily mean that all the sentences will have opinion words in it. Such sentences that do not express any opinion are called objective sentences, while the other sentences in the same document that express some opinion are called subjective sentences. In negation handling task, subjective sentences contribute much in shifting the polarity of the document so it is important to pay additional attention on the subjective sentences [21]. Objective sentences, on the other hand, often have implied negation in it, still attention is mainly on subjective sentences for negation cues and scope modeling.

Attention Vector: Attention mechanism in neural network has ability to focus on subset of the input data. We are representing each word in the sentence $w \in \mathbb{R}^m$ as input vector, $z \in \mathbb{R}^n$ as feature vector, $a \in [0, 1]^n$ as attention vector. Attention mechanism is implemented as

$$a = f_\theta(w)$$
$$g = a \odot z$$

where $f_\theta(w)$ is attention neural network with parameter θ. Attention can be of two types, hard attention, and soft attention. Hard attention has binary value i.e., $a_{hard} = \{0, 1\}^k$ where it completely consider the specific word or completely ignore it. In contrast to hard attention, soft attention $a_{soft} = \mathbb{R}^k$ assigns each word attention weight based on the importance of the word in document.

V. DATA SET

For experiment we used CD dataset annotated with Negation Cues, Scope and Negated Event. The dataset is based on Conan Doyle(CD) stories taken from his three novels. Training dataset(CDT) contains story of The ''Hound of the Baskervilles'', the development corpus(CDD), ''The Adventure of Wisteria Lodge'', and the test corpus(CDE) contains ''The Adventure of the Red Circle'' and ''The Adventure of the Cardboard Box''. Dataset is prepared in CONLL format and which followed the annotation guidelines of BioScope corpus from medical domain. Dataset contains word label information where 1st column is Book Name, 2nd column sentence number, 3rd column contains token/word used in sentence, 4th column contains word, 5th column lemma, 6th column POS (Part of Speech) tag of the word, 7th column Parse Tree information and 8th column contains negation information, which include three components, negation cue, negation scope and negated event. If there are more than one cue in a sentence, then negation information is maintained for each cue with above three components in separate columns. Original text is available in Gutenberg project for research. In annotated dataset if a sentence does not have negation, then the 8th column simply contains ''***'' indicating no cue in sentence.

TABLE 4. Statistics of dataset.

Dataset			
Statistics	**Train(CDT)**	**Dev(CDD)**	**Test(CDE)**
#tokens	65,450	13,566	19,216
#sentences	3644	787	1089
#negationsent.	848	144	235
%negation sent.	23.27	18.29	21.57
#cues	984	173	264
#unique cues	30	20	20
#scopes	887	168	249
#negated	616	122	173

Snapshot of a data from the corpus is shown in below Figure 2 taken from ''baskervilles01'' book from the corpus. The example sentence *''Since we have been so unfortunate as to miss him and have no notion of his errand, this accidental souvenir becomes of importance.''* is 13th sentence inside the book with 25 words shown in 4th column. In this sentence, there are two negation cues ''un'' and ''no'', for which negation ''scope'' and negated event is mentioned in subsequent columns. There is no Scope overlapping for negation cues in the annotated corpus to avoid ambiguity.

VI. METHOD

A. BASE MODEL

For a base line model we used linear SVM classifier for cue identification and probabilistic CRF model for scope detection which served as benchmark for proposed deep neural network model for negation handling.

1) FEATURES FOR NEGATION HANDLING

Machine learning model need to be trained on word level features to learn the syntactic and semantic information. In negation handling tasks, word level features contain information about each word, i.e., how they relate to other words, and what properties they have. There are many word level features that contributes in determining the scope of the negation cue. We need to train the model on input training examples so that model learns the syntactic information about cue and scope. We used Stanford universal dependency parser to generate parse tree to extract syntactic structure of each sentence. During model training it takes following word level features W_{ij}.

$$features(W_{ij}) = \{W_i, L_i, POS_i, Dp_i, W_{i-1}, W_{i+1}, P_i, S_i\}$$

where:

$features(W_{ij})$ is feature vector for word W_i in Sentence S_j.
W_i is i_{th} word or token of sentence S_j.
L_i is lemma of word W_i.

Book Name	Sentence number	Token/ Word number	Token/Word	Lemma	POS tag of the word	Parse tree informatio n	Cue	Scope	Negated Event	Cue	Scope	Negated Event
							Negation information-Cue1			Negation information-Cue2		
baskervilles01	13	0	Since	Since	IN	(S(SBAR*	_	_	_	_	_	_
baskervilles01	13	1	we	we	PRP	(S(NP*)	_	we	_	_	we	_
baskervilles01	13	2	have	have	VBP	(VP(VP*	_	have	_	_	_	_
baskervilles01	13	3	been	be	VBN	(VP*	_	been	_	_	_	_
baskervilles01	13	4	so	so	RB	(ADJP*	_	so	_	_	_	_
baskervilles01	13	5	unfortunate	unfortunate	JJ	*	un	fortunate	fortunate	_	_	_
baskervilles01	13	6	as	as	RB	*	_	as	_	_	_	_
baskervilles01	13	7	to	to	TO	(S(VP*	_	to	_	_	_	_
baskervilles01	13	8	miss	miss	VB	(VP*	_	miss	_	_	_	_
baskervilles01	13	9	him	him	PRP	(NP*))))))	_	him	_	_	_	_
baskervilles01	13	10	and	and	CC	*	_	_	_	_	_	_
baskervilles01	13	11	have	have	VBP	(VP*	_	_	_	_	have	have
baskervilles01	13	12	no	no	DT	(NP(NP*	_	_	_	no	_	_
baskervilles01	13	13	notion	notion	NN	*)	_	_	_	_	notion	_
baskervilles01	13	14	of	of	IN	(PP*	_	_	_	_	of	_
baskervilles01	13	15	his	his	PRP$	(NP*	_	_	_	_	his	_
baskervilles01	13	16	errand	errand	NN	*)))))))	_	_	_	_	errand	_
baskervilles01	13	17	,	,	,	*	_	_	_	_	_	_
baskervilles01	13	18	this	this	DT	(NP*	_	_	_	_	_	_
baskervilles01	13	19	accidental	accidental	JJ	*	_	_	_	_	_	_
baskervilles01	13	20	souvenir	souvenir	NN	*)	_	_	_	_	_	_
baskervilles01	13	21	becomes	become	VBZ	(VP*	_	_	_	_	_	_
baskervilles01	13	22	of	of	IN	(PP*	_	_	_	_	_	_
baskervilles01	13	23	importance	importance	NN	(NP*)))	_	_	_	_	_	_
baskervilles01	13	24	.	.	.	*)	_	_	_	_	_	_

FIGURE 2. Word label feature annotated with negation cue, scope and negated event from CD-SCO corpus.

POS_i is part of speech tag of w_i.
Dp_i is dependency between cue and token w_i.
W_{i-1} is previous word.
W_{i+1} is next word.
P_i is prefix of the word w_i
S_i is suffix of the word w_i

We performed negation handling task is in two phases, identification of all potential negation terms i.e., cues and then determining of part of sentence (scope) impacted by each cue. The task of cue identification is seen as classification problem where we classify words in sentence as potential cue in the input sentence. It is handled as a binary classification problem where model need to classify if a term is negation cue or not. To build feature space for cue classification, we extracted all possible negation cues and their attributes affixes and POS tag of the cue and Cue type, POS tag of next word and previous word, parse tree information, prefix, suffix, Graph distance of token to cue, labeled input training data and stored as a dictionary. To build a baseline model for Cue classification task, we trained *SVM* binary classifier on the training dataset which learn the latent features for negation cues and predict a binary value, 1 for *cue* and 0 for *non_cue* on test dataset. These distinct feature value is treated as a separate attribute for each word in sentences.

In order to train the model on these word features, we converted feature dictionaries into vectors. A binary vector is created for each instance based on the attributes that it contains. To get the single feature vector corresponding to a sentence, we concatenated each feature vector belonging to that sentence them so we end up with a single feature vector for each instance. Once we are done with feature representation part, we trained the model for cue classification task.

In second step we performed scope identification for each cue. For *scope* resolution we used *CRF*, which is a

TABLE 5. Base Model Result for Cue and Scope.

Baseline Model	Precision	Recall	F-measure
Cue Level(SVM)	83.84	79.64	81.68
Scope Level(CRF)	81.26	72.64	76.70

probabilistic model. *CRF* model learn for each term in sentence, if a specific word in sentence falls inside the scope of cue term or not. The model uses BIO labeling to determine span of the cue in sentence. For scope detection we used several token level features as discussed below.

Model's, performance is evaluated on precision, recall and F-Measure values where

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$Fmeasure = (1+\beta^2)\frac{Precision \times Recall}{\beta^2 P + R}$$

We get the base line result as shown in table 5

B. FEATURE ENGINEERING FOR DEEP NEURAL NETWORK

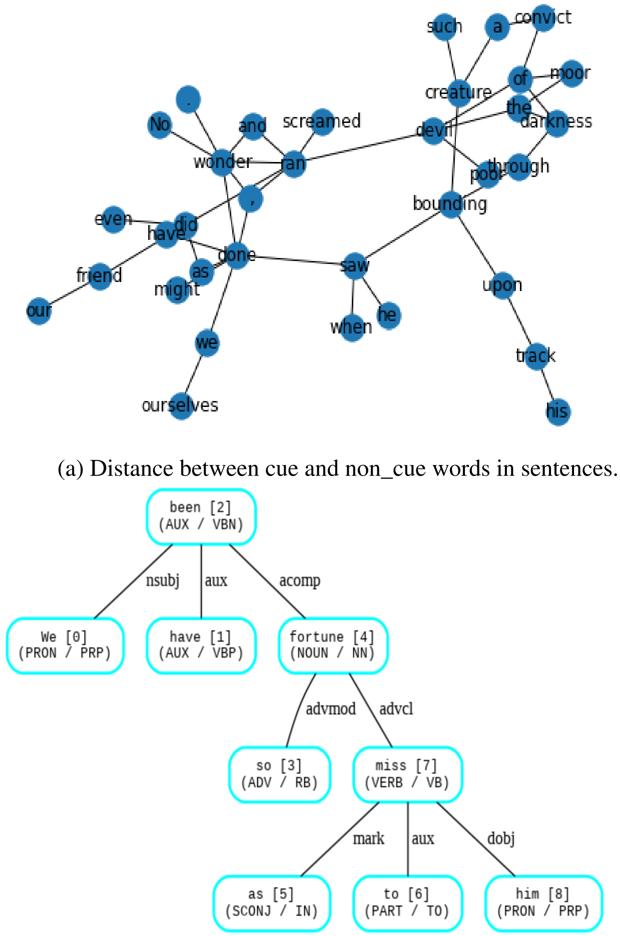
In data preparation stage, we extracted all words from training data which is in CONLL format. These words are represented as a key value pair in dictionary items where key is unique number representing a word and value contains features associated with word in vocabulary. For each word in sentence, We performed feature augmentation to get additional word level features which includes next word, next to next word and lemma, prefix, suffix, POS tag, word lemma, and dependency tree information.

We used both word embedding and character embedding in our model. For Word embedding we used pre-trained word vector model GoogleNewsvectors-negative300.bin.gz to get syntactic and semantic information for each word in sentences. We also used character level embedding to capture morphological negation.

We used networkx python library to use Dijkstra algorithms to find distance between cue and other tokens (see Figure 3) in sentences. To determine dependency relationship among tokens, we generated parse tree using Stanford's universal dependency parser to determine window size where impact of negation cue on other part of sentence can be learned by model. Parse tree for a single example sentence from input dataset is shown in Figure 3b.

The syntactic structure (either constituent or dependency-based) of the sentence is often used to detect the scope of negation. For example, sentence number 13 contains two cues, *"un"* and *"no"*, where scope of cue *"un"* is *"we have been so fortunate as to miss him"* and for other cue *"no"* it is *"have notion of his errand"*. Instead of using phrase structure tree feature from the data set, we converted it into dependency tree to exploit relationship between tokens.

RNN is basically a sequential model. It see input sentences as a sequence of words $S = w_0, w_2, \ldots, w_n$ where each word



(a) Distance between cue and non_cue words in sentences.

(b) Dependency Relationship between cue and non_cue.

FIGURE 3. Deriving Dependency Features using Universal Parse Tree.

w_i appears at a different timestamp t_i. RNN have been mainly used in predicting next word in sequences which has been very successful in real world application. The RNN model accept current input and previous output at any timestamp t_i.

Motivated by the performance of RNN in sequential modeling task, Bi-directional LSTM (B-LSTM), an extension of LSTM, have been used in many sequence tagging applications successfully.

Each input word $w_i in|V|$ is represented as a d dimensional vector. Input feature vector space can be represented as $X \in \mathbb{R}^{t*d}$. We model function as to classify w_i into class c in the output i.e., in our case it is *cue* or *not_cue* $X \in \mathbb{R}^{t*d} \rightarrow \mathbb{R}^c$.

C. CUE AND SCOPE DETECTION

Proposed deep learning linguistic model for negation handling divides the task into two sub-tasks, one as binary classification and the other as sequence labeling sub-task. Model treats cue (negation) identification task as a binary classification problem, where each token in a sentence is classified into cue or non-cue class.

Proposed negation handling model first identifies all cue terms present in each sentence and then determine scope of each cue. Since deep learning model usually works well with large number of features, we performed feature augmentation before model training starts. Each cue term is encoded in binary i.e., labeled 1 for *cue* and 0 for *not_cue*. Input features to consider at word level are prefixes and suffixes, as potential cues. Implied negation cues are often difficult to identify. To identify implied cues, we used similarity score of each token with explicit negation cues. Similarity score for each token w_i is calculated using pre-trained word embedding model which indicate similarity between explicit negation cues and other tokens W_i in sentences. Based on the similarity score between explicit cue and other words, potential implied cue is identified.

VII. ALGORITHM

Steps needed to perform negation handling task is shown in Algorithm 1 which is basically pseudo code that representing end to end process to build a negation handling model. We implemented this algorithm in python using deep learning tensorflow/keras library which require fine grain analysis at each step.

Scope detection task can be treated as sequence labeling problem. Before the rise of neural network based linguistic model, traditional HMM and CRF has been used very frequently in sequence labeling tasks like POS tagging, NER labeling etc. However, with the success of RNN based model in sequence labeling task, we explored it for scope identification for cue in sentence with the target to assign BIO label to each word. BIO label highlights the part of sentence which is impacted by cue term.

It indicates word/token position in sentence i.e., if a word falls at the beginning of a cue's scope, inside the cue's scope or outside it. If there are multiple cue words present in a single sentence, scope is determined for each cue word. Figure4 shows different layers in deep learning model handling specific task at each layer. We used BiLSTM to perform sequence labeling (scope detection) that assigns B-Scope, I-Scope or O-Scope tag to each word in sentence indicating word association with the cue term (see Figure 5). BiLSTM is

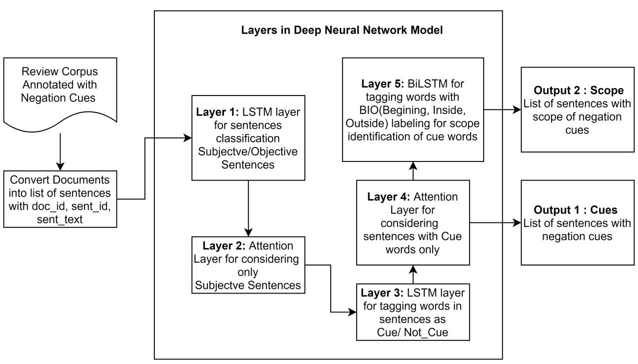


FIGURE 4. **Deep Learning Model for Negation Handling.**

Algorithm 1 Algorithm for Negation Cue and Scope Detection

1: input: text document D, containing ConanDoyle's story.
2: Use NLP sent tokenizer to split document's paragraphs into list of sentences.
$$DocumentD_1 = [s_1, s_2, \ldots, s_n]$$
3: Use word tokenizer on D_1 to represent input document as list of tokens.
$$D_2 = [[w_{11}, \ldots, w_{1i}], [w_{21}, \ldots, w_{2j}], \ldots [w_{n1}, \ldots, w_{nk}]]$$
4: Represent document D_2 as one dimensional vector of words
$$D_2 = (flatten(D_2))$$
$$D_2 = [[w_{11}, \ldots, w_{1i}, [w_{21}, \ldots, w_{2j}, \ldots [w_{n1}, \ldots, w_{nk}$$
5: Get the root word for each token from document D_2 by applying lemmatization.
6: Extract word level features like prefixes, suffixes for each token.
7: Apply dependency parser pipeline to get POS tag and dependency relation among words in the sentences.
8: Perform feature augmentation with explicit negation cue.
$$Neg_{explicit} = [no, not, never, \ldots]$$
9: Use word embedding technique to get token contextual vectors for all tokens in corpus.
10: Calculate similarity score between each word and negation *cues* and extend $Neg_{explicit}$ list with the words having high similarity score.
11: Get vector representation of prefixes and suffixes by applying character embedding on them.
12: Build a sequential neural network model by adding LSTM forward and backward layers, activation function and initialization parameters etc.
13: Store hidden state information until the last timestamp in neural network layer for classification of words as *cue* or *not_cue*
14: Encode positional information into word vectors for identifying scope of the cues.
15: Perform scope resolution of cues by adding another LSTM layer which tags words with B-Scope, I-Scope, and O-Scope indicating span of cue words beginning, inside and outside for a particular cue.

---

extension of LSTM model works in two passes: forward pass and backward pass. It process input sentence word by word from left to right in forward pass and right to left in backward pass. Each input word is an embedded vector representing encoded contextual information. Since word cannot be seen independently in sentence, embedded word vector contains additional information about the neighboring words. In both passes, BiLSTM model with the help of gated mechanism and

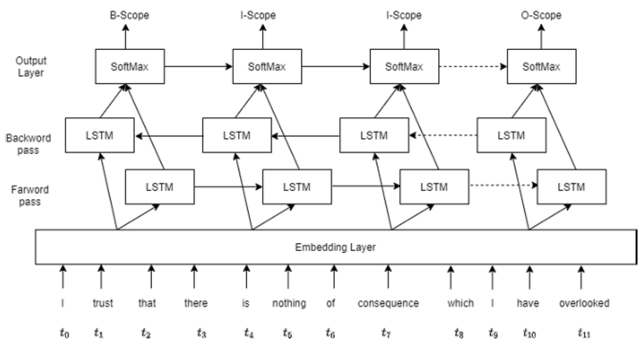


FIGURE 5. BiLSTM for Scope Detection.

cell memory learns the dependency between cue and non_cue words. At the inner layer we used tanh activation function to add non-linearity in the model and at outer layer we used softmax activation function which assigns probability of a word for B-Scope, I-Scope and O-Scope.

VIII. RESULTS AND DISCUSSION

Result shows that deep neural network model performs comparatively better than our baseline linear SVM model for cue classification and probabilistic CRF model for scope identification. We experimented with sequential RNN and its variant, LSTM model with word's positional encoding and embedded feature vector of size 20, 30 and 50 dimensions. Based on the output we can say that LSTM model not only perform better, but also gives flexibility to fine tune the parameters to build a model that solves negation handling problem with minimum human interaction with the feature engineering side. See Table 6 for output achieved with the models. BiLSTM model seems to perform even better with highest F1-Score 93.34%, far ahead of CRF and SVM based models. We implemented code in tensorflow to build deep neural network model for negation cues and scope detection problem. Flexibility with tensorflow in fine tuning model parameters at each layer of the sequential model makes it a good choice to play around with model after pre-processing step. We experimented with many Hyper parameters like learning rate, drop out, activation function selection before coming up with the final result obtained. Few steps in pre-processing we borrowed from existing approach as the data set follows CONLL guidelines.

TABLE 6. Comparative Performance of the Model.

Approach	Precision	Recall	F1-Score
model=CRF	86.65	91.07	88.81
model=SVM	82.44	93.22	87.50
model=RNN	87.84	88.64	88.24
model=LSTM+CRF	85.84	83.64	84.73
model=LSTM + Positional Encoding	89.84	87.64	88.73
model=LSTM+ Embedding	91.84	90.64	91.24
model=BiLSTM+ Embedding	92.86	93.34	93.09

TABLE 7. Model Testing on SFU review Dataset.

LSTM+One Hot Encodings			
Dataset	Precision	Recall	F1-Score
Movies	84.62	81.67	83.12
Books	78.35	80.56	79.44
Cars	78.61	87.34	82.75
Computers	82.65	83.89	83.27
Cookware	81.34	78.21	79.74
Hotels	79.73	75.54	77.58
Music	81.78	77.98	79.83
Phones	82.56	85.87	84.18
LSTM+Word Embedding			
Movies	87.91	86.33	87.11
Books	81.35	85.03	83.15
Cars	81.17	91.54	86.04
Computers	86.91	88.71	87.80
Cookware	85.42	81.45	83.38
Hotels	83.05	78.24	80.57
Music	85.53	81.18	83.30
Phones	84.85	89.81	87.26

For input sentences, model first learns the cues present in the sentences. Negation handling model assigns each word in the sentences a label for both Cues and Scope. For cue labeling, model learns internal state at a time h^t and generate output o^t which in our case is label *in_scope* or *out_scope* at each hidden state with stateful parameter set to true. It then identify scope for each cue within the sentence by assigning B, I and O label for each word in the sentence by using probabilistic softmax activation function the last layer i.e, time distributed dense layer in the model. This BIO label for each word serve as scope for the cue present in sentences. Since model accept input in batches of fixed size, words in one batch may have dependency with words in other batch. LSTM cell state are by default initialize to 0 after each batch is completed and with this initialization network may not have information about the context word which fall into the scope of a word in previous batch. As With stateful parameter of LSTM set to default False value, it only remember what happened within each batch and does not pass information to the next batch. This causes problem as we are note building batches in a way where scope words fall on the boundary of equal length sentences, to overcome this challenge in scope dependency with cue in other batch, we set this flag to True. This way deep neural network model remember all the hidden state and cell states to predict the value at next time sequence.

A. MODEL PARAMETERS

Input text data is converted into three dimensional tensors which is similar to python numpy array, before submitting to the model, where input word to the model is a d dimensional vector of chosen embedding size (see table 8). First dimension of the input shape parameter represents the batch size, second dimension represents the number of time-steps i.e., the number of words given in $max_feature$ size, we are feeding in sequence and the third dimension represents the number of units in one input sequence, in our case it is embedding size. LSTM based language model expect input in word vector format, first we created One Hot Vector from

TABLE 8. Model Hyper Parameter.

Hyper parameters	Value
kernel_initializer	glorot_uniform
recurrent_initializer	orthogonal
bias_initializer	zeros
dropout	0.2
stateful	TRUE
Embedding dimension	20, 30,50
Batch size	100,200,300
Epoch	20

the dataset where each vector is equal to the length of the vocabulary i.e., in our case it was of length 303289 considering the full corpus with only one value corresponding to the index of the word in vocabulary is 1 while rest of the values are 0, resulting in a very large size sparse matrix built over all the words. Other than being sparse, one other issue with this One Hot Encoding representation is words vector is not able to encode any relationships with other words in the sentence, for which we need to define window to capture contextual information. The number of epoch is the total iteration i.e., the number of times model will be trained on the entire dataset. Though choosing smaller batch size will take more time to learn the model each epoch. Since result was almost same after fifteen epochs, we used early stopping callback.

IX. CONCLUSION

Finding an automated solution for linguistic problem is little difficult and this difficulties are inherent. Though Linguistic task does follow some structured grammatical rules, identifying relationship and dependency among tokens to solve even further complex task is a bit challenging. However, based on the experiments and result achieved (see Table 6) we can say that Deep Neural Network based linguistic model performs much better than other state of the art rule based and statistical models. Not only this multi layer Neural Network model performs better, but also model can be generalized to the other domain, as also tested the model on the SFU negation review dataset and result achieved in Table 7 seems satisfactory which could have been even better, had additional pre-processing performed on the dataset. In Neural network based language model size of labeled data significantly matter in fitting the model, not only in learning hidden information, it also ovoid over fitting and under fitting of the model. As the available training data in our case is relatively small, on large input corpus, model is expected to perform even better.

We also tested our model on the SFU dataset and found below results.

Few parameters used during model training is as follow

REFERENCES

[1] L. Jia, C. Yu, and W. Meng, "The effect of negation on sentiment analysis and retrieval effectiveness," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1827–1830.

[2] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo, "A survey on the role of negation in sentiment analysis," in *Proc. Workshop Negation Speculation Natural Lang. Process.*, 2010, pp. 60–68.

[3] F. Benamara, B. Chardon, Y. Mathieu, V. Popescu, and N. Asher, "How do negation and modality impact on opinions?" in *Proc. Workshop Extra-Propositional Aspects Meaning Comput. Linguistics*, 2012, pp. 10–18.

[4] E. Blanco and D. Moldovan, "Semantic representation of negation using focus detection," in *Proc. 49th Annu. Meeting Assoc. Comput. linguistics, Hum. Lang. Technol.*, 2011, pp. 581–589.

[5] S. M. J. Zafra, M. T. M. Valdivia, E. M. Camara, and L. A. U. Lopez, "Studying the scope of negation for Spanish sentiment analysis on Twitter," *IEEE Trans. Affective Comput.*, vol. 10, no. 1, pp. 129–141, Jan./Mar. 2017.

[6] J. S. B. T. Evans, J. Clibbens, and B. Rood, "The role of implicit and explicit negation in conditional reasoning bias," *J. Memory Lang.*, vol. 35, no. 3, pp. 392–409, Jun. 1996.

[7] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *J. Biomed. Informat.*, vol. 34, no. 5, pp. 301–310, 2001.

[8] H. Harkema, J. N. Dowling, T. Thornblade, and W. W. Chapman, "Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports," *J. Biomed. Informat.*, vol. 42, no. 5, pp. 839–851, 2009.

[9] S. Mehrabi, A. Krishnan, S. Sohn, A. M. Roch, H. Schmidt, C. Beesley, M. Schmidt, and M. Palakal, "Deepen: A negation detection system for clinical text incorporating dependency relation into negex," *J. Biomed. Informat.*, vol. 54, pp. 213–219, Oct. 2015.

[10] R. Morante and W. Daelemans, "A metalearning approach to processing the scope of negation," in *Proc. 13th Conf. Comput. Natural Lang. Learn.*, 2009, pp. 21–29.

[11] A. Hogenboom, P. van Iterson, B. Heerschop, F. Frasincar, and U. Kaymak, "Determining negation scope and strength in sentiment analysis," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2011, pp. 2589–2594.

[12] V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik, "The bioscope corpus: Biomedical texts annotated for uncertainty, negation and their scopes," *BMC Bioinf.*, vol. 9, no. 11, pp. 1–9, 2008.

[13] N. Konstantinova, S. C. De Sousa, N. P. C. Díaz, M. J. M. López, M. Taboada, and R. Mitkov, "A review corpus annotated for negation, speculation and their scope," in *Proc. LREC*, 2012, pp. 3190–3195.

[14] R. Morante and W. Daelemans, "Conandoyle-neg: Annotation of negation in conan doyle stories," in *Proc. 8th Int. Conf. Lang. Resour. Eval.*, Istanbul, Turkey, 2012, pp. 1563–1568.

[15] P. G. Mutalik, A. Deshpande, and P. M. Nadkarni, "Use of general-purpose negation detection to augment concept indexing of medical documents: A quantitative study using the UMLS," *J. Amer. Med. Inform. Assoc.*, vol. 8, no. 6, pp. 598–609, Nov. 2001.

[16] P. D. Turney and M. L. Littman, "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," 2002, *arXiv:cs/0212012*. [Online]. Available: https://arxiv.org/ftp/cs/papers/0212/0212012.pdf

[17] S. Sohn, S. Wu, and C. G. Chute, "Dependency parser-based negation detection in clinical narratives," *AMIA Summits Transl. Sci. Proc.*, vol. 2012, p. 1, Oct. 2012.

[18] Y. Ding, "A quantitative analysis of words with implied negation in semantics," *Theory Pract. Lang. Stud.*, vol. 3, no. 7, p. 1200, Jul. 2013.

[19] M. Dadvar, C. Hauff, and F. de Jong, "Scope of negation detection in sentiment analysis," in *Proc. Dutch-Belgian Inf. Retr. Workshop (DIR)*, 2011, pp. 16–20.

[20] U. Farooq, "Negation handling in sentiment analysis at sentence level," *J. Comput.*, vol. 12, pp. 470–478, Sep. 2017.

[21] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," 2004, *arXiv:cs/0409058*. [Online]. Available: https://arxiv.org/abs/cs/0409058

[22] I. Chaturvedi, E. Cambria, R. E. Welsch, and F. Herrera, "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges," *Inf. Fusion*, vol. 44, pp. 65–77, Nov. 2018.

[23] I. Councill, R. McDonald, and L. Velikovich, "What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis," in *Proc. Workshop Negation Speculation Natural Lang. Process.*, 2010, pp. 51–59.

[24] J. Reitan, J. Faret, B. Gambäck, and L. Bungum, "Negation scope detection for Twitter sentiment analysis," in *Proc. 6th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2015, pp. 99–108.

[25] N. P. Cruz, M. Taboada, and R. Mitkov, "A machine-learning approach to negation and speculation detection for sentiment analysis," *J. Assoc. Inf. Sci. Technol.*, vol. 67, no. 9, pp. 2118–2136, Sep. 2016.

[26] M. Enger, E. Velldal, and L. Øvrelid, "An open-source tool for negation detection: A maximum-margin approach," in *Proc. Workshop Comput. Semantics Beyond Events Roles*, 2017, pp. 64–69.

[27] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*. [Online]. Available: http://arxiv.org/abs/1409.3215

[29] L. Lazib, Y. Zhao, B. Qin, and T. Liu, "Negation scope detection with recurrent neural networks models in review texts," *Int. J. High Perform. Comput. Netw.*, vol. 13, no. 2, pp. 211–221, 2019.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: http://arxiv.org/abs/1508.01991

[32] H. Fei, Y. Ren, and D. Ji, "Negation and speculation scope detection using recursive neural conditional random fields," *Neurocomputing*, vol. 374, pp. 22–29, Jan. 2020.

[33] N. Pröllochs, S. Feuerriegel, B. Lutz, and D. Neumann, "Negation scope detection for sentiment analysis: A reinforcement learning framework for replicating human interpretations," *Inf. Sci.*, vol. 536, pp. 205–221, Oct. 2020.

[34] L. T. Slater, W. Bradlow, D. F. Motti, R. Hoehndorf, S. Ball, and G. V. Gkoutos, "A fast, accurate, and generalisable heuristic-based negation detection algorithm for clinical text," *Comput. Biol. Med.*, vol. 130, Mar. 2021, Art. no. 104216.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013, *arXiv:1310.4546*. [Online]. Available: http://arxiv.org/abs/1310.4546

PRAKASH KUMAR SINGH received the B.Tech. degree in information technology from Uttar Pradesh Technical University, Lucknow, India, in 2004, and the M.Tech. degree in information technology from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India, in 2006. He is currently pursuing the Ph.D. degree with the Birla Institute of Technology, Mesra, Ranchi, India.

He has experience of both academic and IT industry. He has experience with many programming languages, tools, and technology related to big data and data science, and currently working with leading IT industry. In 2015, he was working as a Software Engineer at R Systems International Ltd., Noida, India. In 2014, he was working as an Assistant Professor with the Department of Computer Science, Galgotias University, Greater Noida, India. From 2007 to 2013, he was working as an Assistant Professor with the Department of Information Technology, HMR Institute of Technology and Management, New Delhi, India. He is also well familiar and hands on experienced with cloud technology and has been using AWS, Google Cloud, and Azure cloud for five years. He is also working with Spark, Hadoop, Docker, Container, and DevOps technologies. He has strong research interests in the field of machine learning, natural language processing, and deep neural networks.

SANCHITA PAUL received the B.E. degree in computer science and engineering from The University of Burdwan, India, and the M.E. degree in software engineering and the Ph.D. degree in computer science and engineering from the Birla Institute of Technology, Mesra, Ranchi, India. She has total 14 years of experience in teaching and research. Since 2006, she has been working as an Assistant Professor with the Department of Computer Science and Engineering, Birla Institute of Technology, Mesra. She has authored or coauthored 60 scientific publications, including 34 research articles in international journals, 21 articles in international and national conferences, and four book chapters. She has also coauthored a book on green cloud. Her research interests include artificial intelligence, machine learning, natural language processing, and cloud computing. She is a member of the Computer Society of India.