

Deep Semantic Role Labeling for Tweets using 5W1H: *Who, What, When, Where, Why and How*

K. Chakma¹, Amitava Das², Swapan Debbarma²

¹ National Institute of Technology, Agartala,
India

² Indian Institute of Information Technology, Sricity,
India

³ National Institute of Technology, Agartala,
India

kchakma.cse@nita.ac.in, amitava.das@iiits.in, swapanxavier@gmail.com

Abstract. In natural language understanding, Semantic Role Labeling (SRL) is considered as one of the important tasks and widely studied by the research community. State-of-the-art lexical resources have been in existence for defining the semantic role arguments with respect to the predicates. However, such lexical resources are complex in nature which is difficult to understand. Therefore, instead of the classical semantic role arguments, we adopted the concept of 5W1H (*Who, What, When, Where, Why and How*) for SRL. The 5W1H concept is widely used in journalism and it is much simpler and easier to understand as compared to the classical SRL lexical resources. In the recent years, recurrent neural networks (RNN) based end-to-end SRL systems have gained significant attention. However, all recent works have been developed for formal texts. This paper reports on the implementation of a deep neural network using the attention mechanism for extracting the 5W1H from tweets. Our implementation reports an F-1 score of **88.21** which outperforms other recent Twitter SRL system by **28.72**.

Keywords. Semantic role, 5W1H, tweet, attention mechanism.

1 Introduction

Natural language understanding (NLU) is an important and challenging subset of natural language processing (NLP). NLU is considered as

the post-processing of text, after NLP techniques are applied on texts. Semantic Role Labeling (SRL) is a natural language understanding task that extracts semantic constituents of a sentence for answering *who* did *what* to *whom*. SRL is a shallow semantic parsing task whose primary goal is to identify the semantic roles and the relationship among them and therefore, has wide application in other NLP tasks such as Information Extraction [3], Question Answering [33, 21, 9], Machine Translation [16, 36, 38] and Multi-document Abstractive Summarization [10].

The study of semantic roles was first introduced by the Indian grammarian Panini [4] in his “*Karaka*” theory. *Karaka* theory assigns generic semantic roles to words in a natural language sentence. The relationship of the arguments with the verb is described using relations called *Karaka* relations. *Karaka* relations describe the way in which arguments participate in the action described by the verb. Several lexical resources for SRL have been developed such as PropBank [22], FrameNet [2] and VerbNet [31] that define different semantic role sets.

Gildea and Jurafsky [11] developed the first automatic semantic role labeling system based on FrameNet. Subsequent works [26, 27, 23] are considered as traditional approaches that explored

the syntactic features for capturing the overall sentence structure. Most of the SRL works are based on the PropBank [22] role set and use the CoNLL-2005 [5] shared task datasets. These datasets are mainly sections from the World Street Journal (WSJ) articles. Though there have been significant developments in studying SRL, most of the state-of-the-art SRL systems have been developed for formal texts only. However, this paper describes SRL implementation on a different genre of texts called tweets¹.

Twitter is a micro-blogging site that allows a user to post texts (often known as *tweets*) within the limit of 280 characters. Tweets are often found to be informal in nature and tend to be without proper grammatical structures. Use of phonetic typing, abbreviations, word play and emoticons are very common in tweets which therefore, make it a difficult task to perform SRL on such informal texts. Let us illustrate the nature of tweets with some examples.

Examples:

- (1) **Abbreviation:**
 - *IMHO, Elvis is still the king of rock.*
- (2) **Wordplay:**
 - *Sometime things change from **wetoo** to #MeToo.*

In example (1), IMHO is an abbreviation for *in my humble opinion*, whereas in example (2), *wetoo* is the merger of *we* and *too*. In addition to the variations described in examples (1) and (2), users often apply creative writings such as the word *before* is often written as *b4*. These examples suggest that users are at the liberty to write tweets without following the syntactical requirements but maintaining the semantics. Following the above discussions, it appears that performing SRL on tweets is a difficult task. Therefore, the state-of-the-art SRL systems meant for formal texts, are not expected to perform well on tweets. From the available lexical resources for SRL, PropBank is the most commonly studied role set.

¹Texts from www.twitter.com are known as tweets

However, annotations based on the PropBank role set requires sufficient knowledge about the constituent arguments of a predicate. Therefore, instead of using the PropBank role set, we adopted the concept of 5W1H (*Who, What, When, Where, Why, How*) as described in [6]. 5W1H concept is widely used in journalism because an article is considered complete only when all the 5W1H are present. The concept of 5W1H is similar to the *Karaka* relations and easy to understand. We discuss in detail about 5W1H in later sections.

The major contributions of this paper are:

- Development of a corpus for 5W1H extraction from tweets.
- Development of a Deep Neural Network for the 5W1H extraction from tweets.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 discusses the background concepts on SRL. In section 4 the deep neural network implementation is discussed. Section 5 discusses the experiments performed. Results are discussed in section 6 followed by analysis in section 7. We conclude the paper in section 8.

2 Related Work

Though the traditional approaches of Gildea and Jurafsky [11], Pradhan et al. [26], Punyakanok et al. [27] explored the syntactic features, recently, deep neural network based implementations have outperformed the traditional approaches. Zhou and Xu [40] were the first to build an end-to-end system for SRL, where an 8 layered LSTM model was applied which resulted in outperforming the previous state-of-the-art system. To assign semantic labels to syntactic arguments, Roth and Lapata [29] proposed a neural classifier using dependency path embeddings.

He et al. [13] developed a deep neural network with highway LSTMs and constrained decoding that improved over earlier results. To encode syntactic information at word level, Marcheggiani and Titov [19] implemented their system by combining a LSTM network with a graph

convolutional network which improved their LSTM classifier results on the CoNLL-2009 dataset.

Attention mechanism was pioneered by Bahdanau et al. [1]. Cheng et al. used [7] LSTMs and self-attention to facilitate the task of machine reading. Tan et al. [35] implemented a self-attention based neural network for SRL without explicitly modeling any syntax that outperformed the previous state-of-the-art results. Strubell et al. [32] implemented a neural network model that combines multi-head self-attention with multi-task learning across dependency parsing, part-of-speech tagging, predicate detection and SRL. Their [32] method achieved the best scores on the CoNLL-2005 dataset. Liu et al. [17] are the first to study SRL on tweets. They considered only those tweets that reported news events.

They trained a tweet specific system which is based on the mapping between predicate-argument structures from news sentences and news tweets. They further extended their work in [18] where similar tweets are grouped by clustering. Then for each cluster a two-stage SRL labeling is conducted. [20] describe a system for emotion detection from tweets. Their work mainly focuses on identification of roles for *Experiencer*, *State* and *Stimulus* of an emotion. [30] proposed an SRL system for tweets using sequential minimal optimisation (SMO) [25]. Our work adopts the 5W1H extraction for SRL using deep neural network attention mechanism of Bahdanau et al. [1]. Our experiments also show that the attention mechanism is effective on the sequence labeling task of 5W1H.

3 Background

3.1 PropBank based SRL

This section describes the SRL based on the PropBank role set. We first discuss what SRL is and then describe how the PropBank role set is applied for the SRL task. A sentence may represent an event through different surface forms. Let us consider the event of someone (John) *hitting* (event) someone (Steve).

Example:

(3) *Yesterday, John hit Steve with a stick*

The above sentence has different surface level forms such as:

- Steve was hit by John yesterday with a stick
- Yesterday, Steve was hit with a stick by John
- With a stick, John hit Steve yesterday
- Yesterday Steve was hit with a stick by John
- John hit Steve with a stick yesterday

In the above example, despite having different surface level representations, the event is described by the verb (*hit*) where “John” is the syntactic subject and “Steve” is the syntactic object. A subject in a sentence is the causer of the action (verb) whereas, an object is the recipient. From example (3), we are able to represent the fact that there was an event of assault, that the participants in the event are John and Steve, and that John played a specific role, the role of hitting Steve.

These shallow semantic representations are called semantic roles. For a given sentence, the objective of SRL is to first identify the predicates (verb) and the arguments; and classify those arguments of each predicate (verb). In PropBank, every verb (predicate) is described by some senses and for each verb sense, there are specific set of roles defined. For example, the verb *hit* has five different senses in the PropBank database as shown below (only two senses are shown here):

— **sense 1: hit.01, strike**

– **Role:**

- * ARG0: hitter
- * ARG1: thing hit
- * ARG2: instrument hit with

– Example: John hit Steve with a stick

— **sense 2: hit.02, reach, encounter**

– **Role:**

- * ARG0: thing hitting
- * ARG1: thing hit

– Example: Product hit the market

Applying PropBank role set on the sentence in example (3) yields the following semantic roles:

$[Yesterday]_{\text{ARG-TMP}} [John]_{\text{ARG0}} [hit]_{\text{V}}$
 $[Steve]_{\text{ARG1}} [with a stick]_{\text{ARG2}}$.

3.2 Modelling 5W1H

The 5W1H (*Who, What, When, Where, Why and How*) model has been attributed to Hermagoras of Temnos [28] who was an Ancient Greek rhetorician which was further conceptualized by Thomas Wilson [37]. Nowadays, 5W1H is often used in journalism to cover a report. The 5W1H are considered as the answers to a reporter's questions, which are considered as the ground for information gathering. In journalism, a report is considered complete only if it answers to the question of *Who did what, when, where, why and how*.

Let $w = \{w_1, w_2, \dots, w_n\}$ be the sequence of words in a tweet and X be the attribute to which w is to be mapped. We therefore, assume a tweet as $\langle w, X \rangle$, where, X is the tuple $\langle WHO, WHAT, WHEN, WHERE, WHY, HOW \rangle$ in 5W1H.

3.2.1 Defining 5W1H

In this sub section, we define the 5W1H in line with the definitions of [39]. Let $w = \text{"John met her privately, in the hall, on Monday to discuss their relationship"}$:

Definition 1: Who. *It is the set of words that refer to a person, a group of people or an institution involved in an action.*

In w , **Who**={ John }

Definition 2: What. *It is the set of words that refer to the people, things or abstract concepts being affected by an action and which undergo the change of state .*

In w , **What**={ met her }

Definition 3: When. *It is the set of words that refer to temporal characteristics. In tweets, the notion of time may be the days, weeks, months and year of a calendar or the tick of a clock. It also refers to the observations made either before, after or during the occurrence of events such as festivals, ceremonies, elections etc.*

In w , **When**={ on Monday }

Definition 4: Where. *It is the set of the words that refer to locative markers in a tweet. The notion of location is not restricted to physical locations but it also refers to abstract locations.*

In w , **Where**={ in the hall }

Definition 5: Why. *It is the set of words that refer to the cause of an action.*

In w , **Why**={ to discuss their relationship }

Definition 6: How. *It is the set of words that refer to the manner in which an action is performed.*

In w , **How**={ privately }

We denote $\psi_X(w)$ to represent the set of words contained in the text w and classified to the attribute X where, $X \in 5W1H$. According to the Definition 1 to 6, the 5W1H model of tweets can be represented as:

$$\psi_{5W1H}(w) = \bigcup_{X \in 5W1H} \psi_X(w). \quad (1)$$

Table 1. List of PropBank roles

Argument	Role
ARG0	agent
ARG1	patient
ARG2	instrument, benefactive, attribute
ARG3	starting point, benefactive, attribute
ARG4	ending point
ARGM	modifier

3.3 5W1H vs. PropBank

Semantic roles in PropBank are defined with respect to an individual verb sense. In PropBank, the verbs have numbered arguments labeled as: ARG0, ARG1, ARG2, and so on. In general, numbered arguments correspond to the following semantic roles shown in Table 1. Apart from the numbered arguments, PropBank also involves verb modifiers often known as the functional tags such as manner (MNR), locative (LOC), temporal (TMP) and others.

Unlike the PropBank role set, the 5W1H scheme does not specify semantic roles at fine grain levels. However, it defines a simplistic approach for extracting the key information from a given sentence (tweets in our case) for other tasks such as event detection, summerization etc. A comparison of 5W1H and PropBank is illustrated with the following examples.

Example:

(4) *Trump's Pyrrhic Victory Provides a BIG Silver Lining for Democrats* <https://t.co/NzO8NBBkDS>

PropBank on example (4):

— predicate: **provide**

- [Trump's Pyrrhic Victory]_{ARG0}
[Provides]_V [a BIG Silver Lining]_{ARG1}
[for Democrats]_{ARG2}.²

5W1H on example (4):

— predicate: **provide**

²<https://t.co/NzO8NBBkDS>

- [Trump's Pyrrhic Victory]_{Who}
[Provides]_V
[a BIG Silver Lining for Democrats]_{What}³

Example:

(5) *One +ve I will take from Trump's victory is the acknowledged death of political correctness*

PropBank on example (5):

— predicate: **take**

- [One + ve]_{ARG1} [I]_{ARG0}
[will]_{ARGM-MOD} [take]_V
[from Trump's victory]_{ARG2} is the
acknowledged death of political
correctness

— predicate: **acknowledged**

- One +ve I will take from Trump's victory
is the [acknowledged]_V [death]_{ARG1} of
political correctness

5W1H on example (5):

— predicate: **take**

- [One + ve]_{What} [I]_{Who} [will]_{What}
[take]_V [from Trump's victory]_{What}
is the acknowledged death of political
correctness

— predicate: **acknowledged**

- One +ve I will take from Trump's
victory is the [acknowledged]_V
[death of political correctness]_{What}

³<https://t.co/NzO8NBBkDS>

Annotation:

Annotation based on the PropBank role set requires deep knowledge of SRL and the constituent role arguments. On the other hand, the 5W1H annotation scheme is a simplistic Q&A approach as described in our earlier work [6]. Applying the simple question of “*Who did what, when, where, why and how*” yields the constituents of the 5W1H. In example (4), for the predicate **provide**, on applying the 5W1H question of “*Who is the provider*”, yields “*Trump’s Pyrrhic Victory*” as the *Who* and the question “*What is being provided*” yields “*a BIG Silver Lining for Democrats*” as the *What*. Similarly, in example (5), we obtain the 5W1H constituents for each predicate (**take**, **acknowledge**).

However, in both these examples (4 and 5), the arguments ARG1 and ARG2 are merged as “*What*”.

Therefore, the 5W1H model does not distinguish between ARG1 and ARG2. Despite the scenario described in these examples, the constituents of the 5W1H model are mostly similar to some of the PropBank role set. A comparison between the PropBank role set and the 5W1H on our dataset is shown in Table 2. From Table 2, we observe that “*Who*” is mostly similar to ARG0 (84.48%) with a small fraction (10.34%) being similar to ARG1. This is explained with the following example.

Example:

(6) *Murphy Brown Comes Forward With Her Own #MeToo Story* <https://t.co/kKw81IWz5t> via @thedailybeast

In example (6), for the predicate **comes**, “*Murphy Brown* is ARG1 as per PropBank. However, if 5W1H model is applied then “*Murphy Brown* is identified as “*Who*”. An important observation is the coverage of “*What*” with ARG2, ARG4, ARGM-ADV and ARGM-MOD. The 5W1H model does not specify fine grain semantic roles as compared to the PropBank role set. This is already illustrated in examples (4) and (5). From Table 2, we also observe that “*When*”, “*Where*”, “*Why*” and “*How*” are closely similar to ARGM-TMP, ARGM-LOC, ARGM-CAU and ARGM-MNR respectively.

4 Deep Neural Network for 5W1H Extraction

4.1 Attention Background

In this section, we discuss the fundamentals of attention mechanism as proposed by Bahdanau et al. [1]. It is understandable from the example in section 3.1 that SRL is a sequence labeling task. Transforming an input sequence (source) to a new output sequence (target) is the objective of the sequence to sequence (**seq2seq**) model [34]. In **seq2seq**, both sequences can be of arbitrary lengths. The **seq2seq** model is basically an *encoder-decoder* architecture. An *encoder* encodes an input sequence and compresses the information into a context vector of a fixed length. A *decoder* is initialized with the context vector to produce the transformed output.

Under such an architecture, the decoder initial state is obtained only from the last state of the encoder network. However, there is one major disadvantage of this scheme. The fixed-length context vector is incapable of remembering long sentences. In SRL, an argument may span a long sequence in a sentence. In such a scenario, the **seq2seq** model is not suitable because it often forgets earlier parts once it completes processing the whole input. The attention mechanism was introduced by Bahdanau et al. [1] to resolve this problem. Attention mechanism creates connections between the context vector and the entire source input rather than building a single context vector out of the encoder’s last hidden state. The weights of these shortcut connections are customizable for each output element.

A sequence of dense word vectors are used to represent the input sentence. These word vectors are fed to a bi-directional long short-term memory (Bi-LSTM) [14] encoder to produce a series of hidden states that represent the input. Let us consider a source sequence x of length T_x and then use it to output a target sequence y of length T_y :

$$\mathbf{x} = [x_1, x_2, \dots, T_x], \mathbf{y} = [y_1, y_2, \dots, T_y].$$

Table 2. Percentage distribution of similarity between 5W1H and PropBank in our dataset

PropBank Role	Who	What	When	Where	Why	How
ARG0	84.48	0.00	3.33	0.00	0.00	0.00
ARG1	10.34	53.85	0.00	0.00	0.00	0.00
ARG2	0.00	9.89	0.00	0.00	0.00	0.00
ARG3	0.00	0.00	0.00	22.86	0.00	0.00
ARG4	0.00	3.29	0.00	34.29	0.00	0.00
ARGM-TMP	0.00	1.09	60.00	0.00	0.00	0.00
ARGM-LOC	0.00	1.09	10.00	25.71	0.00	0.00
ARGM-CAU	0.00	0.00	0.00	0.00	100.00	0.00
ARGM-ADV	0.00	4.39	20.00	0.00	0.00	0.06
ARGM-MNR	0.00	3.85	0.00	8.57	0.00	90.91
ARGM-MOD	0.00	4.39	0.00	0.00	0.00	0.00
ARGM-DIR	0.00	0.01	0.00	5.71	0.00	3.03
ARGM-DIS	0.00	1.65	0.00	0.00	0.00	0.00
ARGM-NEG	0.00	1.09	0.00	0.00	0.00	0.00

An *encoder* state is represented by the concatenation of the hidden states:

$$\mathbf{h}_i = \left[\begin{array}{c} \rightarrow^T \\ \mathbf{h}_i \\ \leftarrow^T \end{array} \right]^T, \quad i = 1, \dots, T_x. \quad (2)$$

The attention mechanism plugs a context vector c_t between the encoder and the decoder. For each single word that the decoder wants to generate, the context vector is used to compute the probability distribution of source language words .

The context vector c_t depends on a sequence (h_1, \dots, h_{T_x}) to which an encoder maps the input sentence. The decoder has hidden states $s_t = f(s_{t-1}, y_{t-1}, c_t)$ for the output word at position t , $t = 1, \dots, m$, where the context vector c_t is a sum of hidden states of the input sequence, weighted by alignment scores:

$$c_t = \sum_{i=1}^n \alpha_{t,i} \mathbf{h}_i, \quad (3)$$

$$\alpha_{t,i} = \frac{\exp(\text{score}(s_{t-1}, \mathbf{h}_i))}{\sum_{i'=1}^n \exp(\text{score}(s_{t-1}, \mathbf{h}_{i'}))}, \quad (4)$$

$\alpha_{t,i}$ is the alignment score assigned to the pair of input at position i and output at position t , (y_t, x_i) which depends on the proximity of their match. The extent of each source hidden state to be chosen for each output depends on the set of weights

$\{\alpha_{t,i}\}$. To parameterize the alignment score α , a feed-forward network with a single hidden layer is used and this network is jointly trained with other parts of the model. Therefore, the score function is represented in the following form, given that *tanh* is used as the non-linear activation function:

$$\text{score}(s_t, \mathbf{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[s_t; \mathbf{h}_i]), \quad (5)$$

where both \mathbf{v}_a and \mathbf{W}_a are weight matrices to be learned in the alignment model.

4.2 Architecture

Our deep attention architecture is in similar lines with Bahdanau et. al. [1] where the input is a sequence of words $(x_1, x_2, \dots, x_{T_x})$ consisting of source word tokens and 5W1H tokens. The input sequence are mapped to the target sequence $(y_1, y_2, \dots, y_{T_y})$ by our trained model(s). The source sentence is represented as dense word vectors which is then fed to a Bi-directional LSTM encoder to generate the hidden states as stated in eq.(2). The information generated by the hidden states is then used by the decoder to output the target tokens recursively. The architecture of the deep neural network is shown in Fig 1.

4.2.1 Encoder

Our encoder is a B-directional RNNs (Recurrent Neural Network) with LSTM cells. The encoder

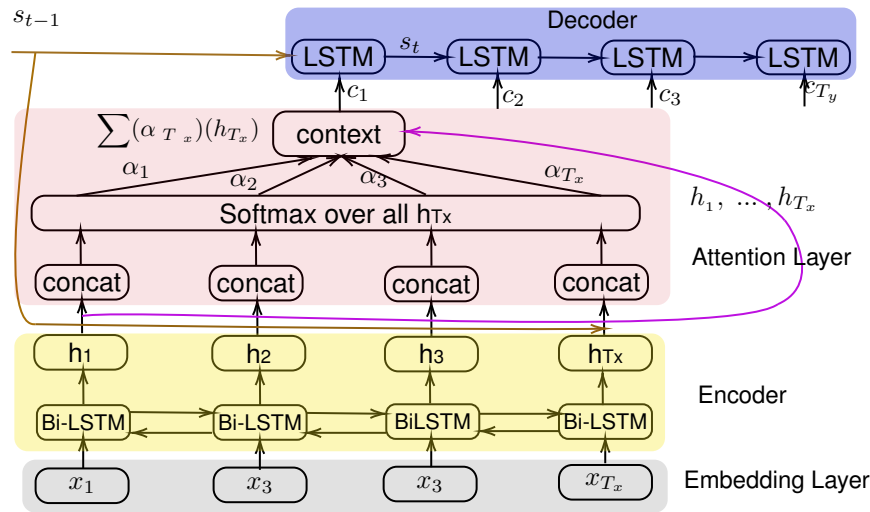


Fig. 1. Deep Neural Attention Network

outputs hidden states $\left[\begin{matrix} \rightarrow \\ h_i \\ \leftarrow \\ h_i \end{matrix} \right]$ where each h_i contains information about the surrounding context of the word x_i . We refer to \mathbf{M} as the complete matrix of encoder hidden states. Since the length of the input sequence is T_x , the shape of the output of the encoder is $T_x \times 2\mathbf{M}$.

4.2.2 Attention

The attention mechanism is a feed-forward neural network of two layers. In the first layer, at every time step t , we use the concatenation of the forward and backward source hidden states h in the bi-directional encoder and target hidden states s_{t-1} in the previous time step of the non-stacking unidirectional decoder. The score $score(s_t, h_i)$ in eq.(5) is the result of the concatenation which is then passed to a softmax to generate the $\alpha_{t,i}$. Since the $\alpha_{t,i}$ is generated for only one hidden state h_i , we need to apply the softmax over all the h of the input sequence T_x . This is obtained by copying the s_{t-1} to all the h in T_x :

$$\begin{aligned} \text{out}(1) &= \text{NeuralNet}([s(t-1), h(1)]), \\ \text{out}(2) &= \text{NeuralNet}([s(t-1), h(2)]), \\ &\dots \\ \text{out}(T_x) &= \text{NeuralNet}([s(t-1), h(T_x)]), \end{aligned}$$

Therefore, $\alpha_{t,i}$ is calculated as:

$$\alpha_{t,i} = \frac{\exp(\text{score}(s_{t-1}, h_i))}{\sum_{i=1}^{T_x} \exp(\text{score}(s_{t-1}, h_{T_x}))}. \quad (6)$$

The generated $\alpha_{t,i}$ are then used with the hidden states h in T_x to compute the context vector c_t which is a weighted sum of the products of $\alpha_{t,i}$ and h_i in T_x as shown in eq.(3).

4.2.3 Decoder

The decoder is a single layer unidirectional LSTM network responsible for generating the output token y_t where $t = 1, 2, \dots, T_y$. A learned distribution over the vocabulary at each time step is used to generate y_t from t given its state s_t , the previous output token y_{t-1} , the context vector c_t and \mathbf{M} . We can parameterize the probability of decoding each word yt as:

$$p(y_t | s_t, y_{t-1}, c_t, \mathbf{M}) = \text{softmax}(g(s_t)), \quad (7)$$

where g is the transformation function that outputs a vocabulary-sized vector. Here, s_t is the RNN hidden unit, abstractly computed as:

$$s_t = f(s_{t-1}, h), \quad (8)$$

where f computes the current hidden state given the previous hidden state in a LSTM unit.

Minimization of the negative log-likelihood of the target token y_t for each time step is the primary objective of our model at the time of training. The loss for the whole sequence (X) is calculated as:

$$loss = -\frac{1}{T_y} \sum_{t=0}^{T_y} \log P(y_t | y < t, X). \quad (9)$$

5 Experiments

5.1 Dataset

We used two different datasets, one based on the US Elections held in November, 2016 and the other based on the #MeToo⁴ campaign. The dataset on the US Elections are taken from [30] containing 3000 English tweets. For the #MeToo dataset, we crawled 248,160 tweets using hash tags such as #MeToo, #MeTooCampaign, #MeTooControversy, #MeTooIndia, as query with the *twitter4j*⁵ API. We applied regular expressions to remove the Re-tweets (tweets with RT prefix) and Non-English tweets. Most of the Non-English tweets are in Roman transliterated form and therefore, they are manually removed.

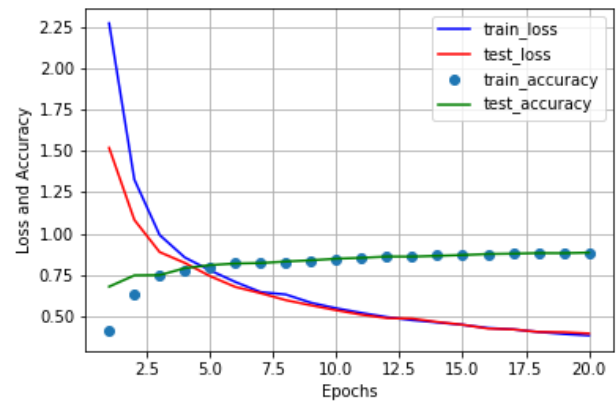
After manually removing the re-tweets and Non-English tweets, the dataset is finally reduced to 8175 tweets. The reason for such a drastic reduction is due to the presence of tweets containing Roman transliterated Non-English words. All the tweets are then tokenized with CMU tokenizer [12]. We prepared the datasets in such a manner that for every tweet that has multiple predicates, the tweet is repeated in the corpus for each predicate (Table 3).

5.2 Model Setup

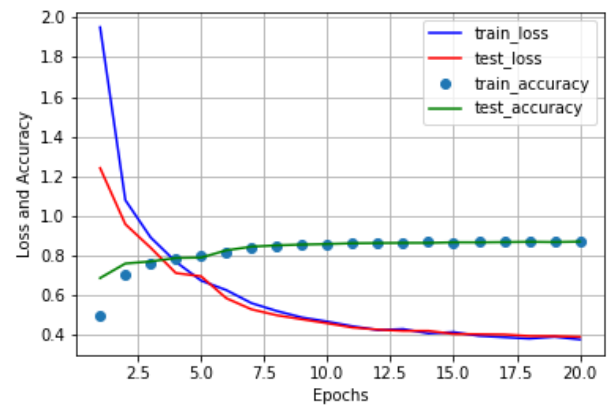
We setup our model with Keras [8] and initialize the model with pre-trained 300-dimensional GloVe [24] embeddings. Our vocabulary size is set to $|\nu| \approx 40K$ words with a maximum sequence length = 100. Both the *encoder* and *decoder* are set with latent dimensions of 256 respectively. For the *attention* layer, we use Keras RepeatVector [8]

⁴[https://en.wikipedia.org/wiki/Me_Too_movement_\(India\)](https://en.wikipedia.org/wiki/Me_Too_movement_(India))

⁵<http://twitter4j.org/en/index.html>



(a) Loss and Accuracy on the US elections dataset



(b) Loss and Accuracy on the #MeToo dataset

Fig. 2. Reported loss and accuracy

set to the maximum length of the input sequence. The concatenation $[s_{t-1}, h]$ is obtained by merging them into a concatenated layer. The *attention* layer is implemented with two *dense* layers set with *tanh* and *softmax* respectively. The softmax is applied across all the hidden states from the encoder. For obtaining the context c_t , at every time step t , a different layer is required. We implemented this using Lambda layer wrappers of the Keras API.

5.3 Learning

Our models are created on a fifth generation Intel core i7 based machine with four cores and 16

Table 3. Dataset structure. Every tweet that has n predicates is repeated n times in the dataset

Repeat count	predicate	Tweet
1	attempt	[Apple CEO Tim Cook] _{WHO} [attempts] _v [to unify staff in wake of Trump victory] _{WHAT}
2	unify	[Apple CEO Tim Cook] _{Who} attempts to [unify] _v [staff in wake of Trump victory] _{WHAT}

gigabytes of Random Access Memory without any GPU (Graphics Processor Unit) support. Due to the lack of a GPU support, a single epoch takes a considerable amount of time. With the available system configuration, a single epoch took around 30 to 40 minutes. We therefore, experimented with only different epochs of 5, 10 and 20 and got the best results with 20 epochs with a batch size, $BATCH_SIZE = 1000$. We use Adam optimizer [15] and a learning rate $l_r = 0.1$. The dataset was split into 90% train and 10% test sets.

6 Results

The objective of our work is to extract the 5W1H from tweets. But for comparison with previous [30] SRL systems on tweets, we evaluated our system (*Deep-SRL*) for PropBank role identification task. In Table 4, we give the comparison of our system (*Deep-SRL*) with the SRL system of Rudrapal et. al. [30] (*DRP-SRL*) on the PropBank role identification task. For evaluation purpose, we used the standard measures of *Precision*, *Recall* and *F-1*.

The comparison is done on the *US Elections 2016* dataset, on which our system outperformed *DRP-SRL* system by overall F-1 of 28.72. This is a significant improvement over previous results. In Table 5, we give the performance of *Deep-SRL* for 5W1H extraction on both the two datasets (*US Elections 2016* and *metoo movement*). *Deep-SRL* achieves an overall F-1 score of 88.21 in the whole corpus.

Fig 2(a) and (b) show the loss and accuracy of our model on the train and test sets on both the datasets respectively. The reported loss at *epoch* = 20 for the *US Elections* dataset is 0.5 and that for

the *#MeToo* dataset is 0.45. This indicates a drop in the loss by 0.05.

Our models reported an accuracy of 88.32% for the *US Elections* dataset and 88.15% for the *#MeToo* dataset. The three metrics of precision, recall and F-1 score is shown in fig 3(a) and (b).

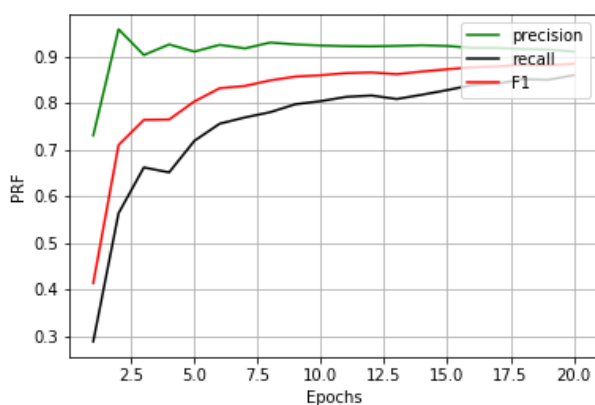
7 Analysis

Since we adopted the BIO⁶ tagging format, it is necessary to identify the **argument span**. Here, argument span means the maximum number of tokens falling under a WHO or WHAT or WHEN or WHERE or WHY or HOW. To verify argument spans, we measure the percentage of overlap between the predicted argument spans and the gold spans. We found that 85.4% of the predicted spans match the gold spans completely, 5.23% of the predicted spans are partially overlapping with gold spans, and 9.37% of the predicted spans do not overlap at all with gold. There are partial overlaps because the model could not tag some of the group of tokens with a proper BIO sequence. For example, a token which is supposed to be tagged with a B-WHO, was tagged as I-WHO.

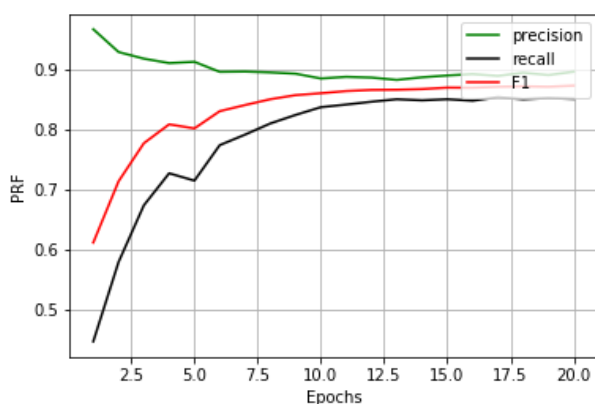
Table 4. Comparison of *DRP* and our system on the PropBank role identification task for the *US Election* corpus.

System	#Tweets	F-1
DRP-SRL	3000	59.76
Deep-SRL	3000	88.48

⁶[https://en.wikipedia.org/wiki/Inside%E2%80%93beginning_\(tagging\)](https://en.wikipedia.org/wiki/Inside%E2%80%93beginning_(tagging))



(a) Precision, Recall and F-1 on the US elections dataset



(b) Precision, Recall and F-1 on the #MeToo dataset

Fig. 3. Model performance on all the datasets

8 Conclusion

SRL based on the PropBank role set is a fine-grained approach but requires deep knowledge about the role arguments for annotation. In contrast, our simplistic 5W1H approach is easier to annotate a corpus with a little compromise at the fine-grain level role set identification task. In this work, we proposed a deep attention based neural network for the task of semantic role labeling by extracting the 5W1H from tweets. We trained our models and evaluated them on the 2016 US Elections dataset that was used by a previous SRL system for tweets. We compared

Table 5. Our System (Deep-SRL) for 5W1H extraction on both the US Election and #MeToo corpus.

Corpus	Precision	Recall	F-1
US Elections	90.87	86.21	88.48
#MeToo	90.63	85.40	87.94
Average	90.75	85.8	88.21

our models with previous SRL systems on tweets and observed a significant improvement over the previous implementations. We also prepared a new dataset based on the #MeToo campaign and evaluated our models on them. Our experimental results indicate that our models substantially improve SRL performances on tweets. However, there are certain limitations in the 5W1H adoption as the fine-grain semantic roles are ignored in such an approach, thus, limiting the in-depth SRL role identification. However, the 5W1H concept could be very convenient to perform other information extraction tasks such as event detection, even summarization on tweets.

References

1. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *Proceedings of the ICLR Conference.*, ICLR, pp. .
2. Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet project. *Association for Computational Linguistics '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Vol. 1, pp. 86–90.
3. Bastianelli, E., Castellucci, G., Croce, D., & Basili, R. (2013). Textual inference and meaning representation in human robot interaction. *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, JSSP.
4. Bharati, A., Chaitanya, V., & Sangal, R. (1994). *Natural Language Processing a Paninian Perspective*, volume 1. Prentice-Hall, PHI, New Delhi.

5. **Carreras, X. & Màrquez, L. (2005).** Introduction to the conll-2005 shared task: semantic role labeling. *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, pp. 152–164.
6. **Chakma, K. & Das, A. (2018).** A 5w1h based annotation scheme for semantic role labeling of english tweets. *Computación y Sistemas*, Vol. 22, No. 3, pp. 747–755.
7. **Cheng, J., Dong, L., & Lapata, M. (2016).** Long short-term memory-networks for machine reading. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 551–561.
8. **Chollet, F. et al. (2015).** Keras. <https://keras.io>.
9. **Dan, S. & Lapata, M. (2007).** Using semantic roles to improve question answering. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, pp. 12–21.
10. **Genest, P. & Lapalme, G. (2011).** Framework for abstractive summarization using text-to-text generation. *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, Association for Computational Linguistics, pp. 64–73.
11. **Gildea, D. & Jurafsky, D. (2002).** Automatic labeling of semantic roles. *Association for Computational Linguistics*, Vol. 28, No. 3, pp. 245–288.
12. **Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., & Smith, N. A. (2011).** Part-of-speech tagging for twitter: Annotation, features, and experiments. *Proceedings of Association for Computational Linguistics 2011*, Association for Computational Linguistics, pp. 42–47.
13. **He, L., Lee, K., Lewis, M., & Zettlemoyer, L. S. (2017).** Deep semantic role labeling: What works and what's next. *Association for Computational Linguistics*, pp. 473–483.
14. **Hochreiter, S. & Schmidhuber, J. (1997).** Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780.
15. **Kingma, D. P. & Ba, J. (2014).** Adam: A method for stochastic optimization. *CoRR*, Vol. abs/1412.6980.
16. **Knight, K. & Luk, S. K. (1994).** Building a large-scale knowledge base for machine translation. *AAAI*, American Association for Artificial Intelligence, pp. 773–778.
17. **Liu, X., Li, K., Han, B., Zhou, M., Jiang, L., Xiong, Z., & Huang, C. (2010).** Semantic role labeling for news tweets. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Association for Computational Linguistics, pp. 698–706.
18. **Liu, X., Li, K., Han, B., Zhou, M., & Xiong, Z. (2011).** Collective semantic role labeling for tweets with clustering. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI'11)*, Vol. 3, pp. 1832–1837.
19. **Marcheggiani, D. & Titov, I. (2017).** Encoding sentences with graph convolutional networks for semantic role labeling. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 1506–1515.
20. **Mohammad, S. M., Zhu, X., & Martin, J. (2010).** Semantic role labeling of emotions in tweets. *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 32–41.
21. **Moschitti, A., Morarescu, P., & Harabagiu, S. M. (2003).** Open domain information extraction via automatic semantic labeling. *FLAIRS Conference*, pp. 397–401.
22. **Palmer, M., Gildea, D., & Kingsbury, P. (2005).** The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics Journal*, Vol. 31, No. 1, pp. 71–105.
23. **Palmer, M., Gildea, D., & Xue, N. (2010).** Semantic role labeling. *Synthesis Lectures on Human Language Technology Series*, Morgan and Claypool.
24. **Pennington, J., Socher, R., & Manning, C. D. (2014).** Glove: Global vectors for word representation. *EMNLP*, pp. 1532–1543.
25. **Platt, J. C. (1998).** Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING.
26. **Pradhan, S., Haciogl, K., Ward, W., Martin, K., & Jurafsky, D. (2005).** Semantic role chunking combining complimentary syntactic views. *Proceedings of the 9th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, pp. 217–220.

27. **Punyakanok, V., Roth, D., & Yih, W.-T. (2005).** The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, Association for Computational Linguistics, pp. 257–287.
28. **Robertson, D. & Bruce, W. (1946).** A note on the classical origin of 'circumstances' in the medieval confessional. *Studies in Philology*, Vol. 43, No. 1, pp. 6–14.
29. **Roth, M. & Lapata, M. (2016).** Neural semantic role labeling with dependency path embeddings. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 1192–1202.
30. **Rudrapal, D. & Das, A. (2018).** Semantic role labelling of english tweets through sentence boundary detection. *International Journal of Intelligent Information and Database Systems*, Vol. 11, No. 4, pp. 225–235.
31. **Schuler, K. K. & Palmer, M. S. (2005).** *Verbnet: a broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania Philadelphia, PA, USA, Philadelphia, USA.
32. **Strubell, E., Verga, P., Andor, D., Weiss, D., & McCallum, A. (2018).** Linguistically-informed self-attention for semantic role labeling. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 5027–5038.
33. **Surdeanu, M., Harabagiu, S., Williams, J., & Aarseth, P. (2003).** Using predicate-argument structures for information extraction. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, pp. 8–15.
34. **Sutskever, I., Vinyals, O., Le, Q. V., & McCallum, A. (2014).** Sequence to sequence learning with neural networks. *Proceedings of NIPS*, MIT Press, pp. 3104–3112.
35. **Tan, Z., Wang, M., Xie, J., Chen, Y., & Shi, X. (2018).** Deep semantic role labeling with self-attention. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pp. 4929–4936.
36. **Ueffing, N., Haffari, G., & Sarkar, A. (2007).** Transductive learning for statistical machine translation. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Association of Computational Linguistics, pp. 25–32.
37. **Wilson, T. & Medine, P. (1994).** *The Art of Rhetoric (1560)*. G - Reference, Information and Interdisciplinary Subjects Series. Pennsylvania State University Press.
38. **Wu, D. & Fung, P. (2009).** Semantic roles for smt: a hybrid two-pass model. *Proceedings of Human Language Technologies*, Association for Computational Linguistics, pp. 13–16.
39. **Zhao, Z., Sun, J., Mao, Z., Feng, S., & Bao, Y. (2016).** Determining the topic hashtags for Chinese microblogs based on 5W model. *Proc. of the second International Conference, BigCom 2016*, Springer, pp. 55–67.
40. **Zhou, J. & Xu, W. (2015).** End-to-end learning of semantic role labeling using recurrent neural networks. *Proc. of the Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 1127–1137.

Article received on 23/01/2019; accepted on 04/03/2019.
Corresponding author is K. Chakma.