

AMERICAN UNIVERSITY OF BEIRUT

Extracting War Incidents from News Articles via  
Deep Sequence Tagging

by

Nancy Joseph Sawaya

A thesis

submitted in partial fulfillment of the requirements  
for the degree of Master of Science  
to the Department of Computer Science  
of the Faculty of Arts and Sciences  
at the American University of Beirut

Beirut, Lebanon  
September 2019

# AMERICAN UNIVERSITY OF BEIRUT

## Extracting War Incidents from News Articles via Deep Sequence Tagging

by  
Nancy Joseph Sawaya

Approved by:

---

Dr. Shady Elbassuoni, Assistant Professor  
Computer Science

Advisor



---

Dr. Fatima Abu Salem, Associate Professor  
Computer Science

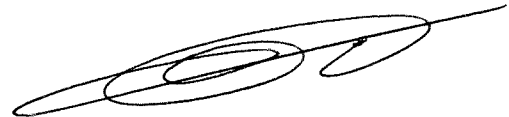
Member of Committee



---

Dr. Mohamed El Baker Nassar, Assistant Professor  
Computer Science

Member of Committee



Date of thesis defense: September 3, 2019

# AMERICAN UNIVERSITY OF BEIRUT

## THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Sawaya

Nancy

Joseph

---

Last

First

Middle

Master's Thesis

Master's Project

Doctoral Dissertation

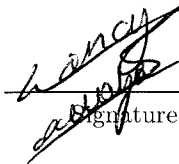
I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after:

One \_\_\_ year from the date of submission of my thesis, dissertation or project.

Two \_\_\_ years from the date of submission of my thesis , dissertation or project.

Three  years from the date of submission of my thesis , dissertation or project.

  
Signature

6/9/2019  
Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

# Acknowledgements

I would like to share with you my deepest appreciation to my thesis advisor Prof. Shady Elbassuoni for his professional guidance and assistance. His constructive comments have encouraged me to work on my thesis unstoppably. I would also like to thank the committee member Prof. Fatima Abu Salem who made a huge contribution to the thesis throughout the process. Without their support, I would have never finished this work.

I would also like to show gratitude to Prof. Mohamed El Baker Nassar who accepted to serve on my committee. He raised many valuable points that helped in improving my work.

I take this opportunity to thank my University, the American University of Beirut where I have met amazing professors.

Lastly, I would like to thank my friends Roula and Samar for their endless support during my studies.

# An Abstract of the Thesis of

Nancy Joseph Sawaya for Master of Science  
Major: Computer Science

Title: Extracting War Incidents from News Articles via Deep Sequence Tagging

An important natural language processing (NLP) task is to extract structured information from free text. In this thesis, we focus on the problem of extracting war incidents from news articles. A war incident is a tuple consisting of a location of the incident, the actor, the cause of death, and the number of casualties. We employ OpenTag [1], a deep sequence-tagging approach, followed by a series of flat classifiers to achieve this task. To train our sequence tagging model and the flat classifiers, we utilize a dataset of news articles surrounding the Syrian war. Our approach, which utilizes sequence tagging, outperforms baseline classifiers that rely solely on the text of the news articles.

# Contents

<b>Acknowledgements</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Sequence Tagging . . . . .	2
1.3 Recurrent Neural Networks . . . . .	3
1.4 Classifiers . . . . .	4
1.5 Objectives and Contributions . . . . .	4
1.6 Thesis Plan . . . . .	5
<b>2 Joplin and FA-KES Challenges</b>	<b>7</b>
<b>3 Literature Review</b>	<b>10</b>
3.1 Natural Language Processing . . . . .	10
3.2 Open Attribute Value Extraction . . . . .	13

<b>4</b>	<b>Dataset</b>	<b>15</b>
4.1	Joplin Dataset . . . . .	15
4.2	FA-KES Dataset . . . . .	19
<b>5</b>	<b>Proposed Approach</b>	<b>23</b>
5.1	Joplin Dataset . . . . .	23
5.1.1	Majority Vote . . . . .	23
5.2	FA-KES Dataset . . . . .	24
5.2.1	Answer Retrieval . . . . .	24
5.3	Sequence Tagging . . . . .	25
5.4	Deep Learning Model . . . . .	32
5.5	FA-KES Classifiers . . . . .	34
5.5.1	Actor Classifier . . . . .	36
5.5.2	Location Classifier . . . . .	37
5.5.3	Cause of Death Classifier . . . . .	38
5.5.4	Civilians Classifier . . . . .	39
<b>6</b>	<b>Metrics</b>	<b>40</b>
<b>7</b>	<b>Evaluations</b>	<b>45</b>
7.1	Evaluation of Deep Learning on Joplin Dataset . . . . .	45
7.1.1	Casualties and Damages Dataset . . . . .	46
7.1.2	Caution and Advice Dataset . . . . .	48

7.1.3	Donations Dataset . . . . .	50
7.1.4	Information Source . . . . .	52
7.1.5	Discussion . . . . .	54
7.1.6	Error Analysis . . . . .	59
7.2	Evaluation of Deep Learning on FA-KES Dataset . . . . .	70
7.2.1	FA-KES Titles Dataset . . . . .	71
7.2.2	FA-KES Titles and First Paragraphs Dataset . . . . .	78
7.2.3	FA-KES Titles and Contents Dataset . . . . .	80
7.2.4	Discussion . . . . .	82
7.2.5	Error Analysis . . . . .	86
7.3	Evaluation of FA-KES Classifiers . . . . .	93
7.3.1	Actor Classifier . . . . .	93
7.3.2	Location Classifier . . . . .	99
7.3.3	Cause of Death Classifier . . . . .	106
7.3.4	Civilians Classifier . . . . .	111
7.4	Discussion . . . . .	114
<b>8</b>	<b>Conclusion</b>	<b>118</b>
<b>A</b>	<b>Abbreviations</b>	<b>120</b>
	<b>Bibliography</b>	<b>122</b>



# List of Figures

7.1	CRF loss on Casualties and Damages Training and Validation Datasets . . . . .	48
7.2	BiLSTM-CRF loss on Casualties and Damages Training and Validation Datasets . . . . .	48
7.3	CRF loss on Caution and Advice Training and Validation Datasets	49
7.4	BiLSTM-CRF loss on Caution and Advice Training and Validation Datasets . . . . .	49
7.5	CRF loss on Donations Training and Validation Datasets . . . . .	51
7.6	BiLSTM-CRF loss on Donations Training and Validation Datasets	51
7.7	CRF loss on Information Source Training and Validation Datasets	53
7.8	BiLSTM-CRF loss on Information Source Training and Validation Datasets . . . . .	53
7.9	CRF loss on FA-KES Titles Training and Validation Datasets . .	71
7.10	OpenTag loss on FA-KES Titles Training and Validation Datasets	71
7.11	Example of Attention Visualization on FA-KES Titles testing dataset	72

7.12	Example of Attention Visualization on FA-KES Titles testing dataset	72
7.13	Example of Attention Visualization on FA-KES Titles testing dataset	72
7.14	Example of Attention Visualization on FA-KES Titles testing dataset	72
7.15	CRF loss on FA-KES Titles and First Paragraphs Training and Validation Datasets . . . . .	78
7.16	OpenTag loss on FA-KES Titles and First Paragraphs Training and Validation Datasets . . . . .	78
7.17	CRF loss on FA-KES Titles and Contents Training and Validation Datasets . . . . .	81
7.18	OpenTag loss on FA-KES Titles and Contents Training and Vali- dation Datasets . . . . .	81

# List of Tables

4.1	Sample of Casualties and Damages’ tweets . . . . .	16
4.2	Sample of Caution and Advice tweets . . . . .	17
4.3	Sample of Donations’ tweets . . . . .	18
4.4	Sample of Information Source tweets . . . . .	20
4.5	Sample of FA-KES articles . . . . .	22
5.1	Answers extracted from FA-KES’ article shown in Table 4.5 . . . . .	26
5.2	Example of Casualties and Damages’ tweet . . . . .	28
5.3	Example of Caution and Advice tweet . . . . .	29
5.4	Example of Donations’ tweet . . . . .	30
5.5	Example of Information Source tweet . . . . .	31
5.6	Example of FA-KES’ articles . . . . .	32
6.1	“Four Outcomes of Binary Classification” taken from [2] . . . . .	44
7.1	Results of CRF and BiLSTM-CRF on Casualties and Damages testing dataset . . . . .	47

7.2	Classification report of CRF model on Casualties and Damages testing dataset . . . . .	47
7.3	Classification report of BiLSTM-CRF on Casualties and Damages testing dataset . . . . .	47
7.4	Results of CRF and BiLSTM-CRF on Caution and Advice testing dataset . . . . .	49
7.5	Classification report of the CRF model on Caution and Advice testing dataset . . . . .	50
7.6	Classification report of BiLSTM-CRF on Caution and Advice testing dataset . . . . .	50
7.7	Results of CRF and BiLSTM-CRF on Donations testing dataset	51
7.8	Classification report of the CRF model on Donations testing dataset	52
7.9	Classification report of BiLSTM-CRF on Donations testing dataset	52
7.10	Results of CRF and BiLSTM-CRF on Information Source testing dataset . . . . .	53
7.11	Classification report of the CRF model on Information Source testing dataset . . . . .	53
7.12	Classification report of BiLSTM-CRF on Information Source testing dataset . . . . .	54
7.13	Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Casualties and Damages testing dataset . . . . .	60

7.14	Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Casualties and Damages testing dataset . . . . .	61
7.15	Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Caution and Advice testing dataset . . . . .	64
7.16	Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Donations testing dataset . . . . .	67
7.17	Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Information Source testing dataset . . . . .	69
7.18	Sample of Information Source tweets Extracted Information . . .	70
7.19	Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.11 . . . . .	73
7.20	Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.12 . . . . .	74
7.21	Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.13 . . . . .	75
7.22	Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.14 . . . . .	76
7.23	Classification report of the CRF model on FA-KES Titles testing dataset . . . . .	77
7.24	Classification report of OpenTag on FA-KES Titles testing dataset	77
7.25	Classification report of the CRF model on FA-KES Titles and First Paragraphs testing dataset . . . . .	79

7.26	Classification report of OpenTag on FA-KES Titles and First Paragraphs testing dataset . . . . .	80
7.27	Classification report of the CRF model on FA-KES Titles and Contents testing dataset . . . . .	81
7.28	Classification report of OpenTag on FA-KES Titles and Contents testing dataset . . . . .	82
7.29	Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles testing dataset . . . . .	89
7.30	Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles testing dataset . . . . .	90
7.31	Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles and First Paragraphs testing dataset . . . . .	91
7.32	Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles and Contents testing dataset . . . . .	92
7.33	Classification report of the baseline Logistic Regression Actor classifier on FA-KES Titles testing dataset . . . . .	94
7.34	Classification report of Logistic Regression Actor classifier on FA-KES Titles testing dataset with tags predicted by OpenTag . . . . .	95
7.35	Classification report of the baseline Logistic Regression Actor classifier on FA-KES Titles and First Paragraphs testing dataset . . . . .	96

7.36	Classification report of Logistic Regression Actor classifier on FA-KES Titles and First Paragraphs testing dataset with tags predicted by OpenTag . . . . .	97
7.37	Classification report of the baseline Logistic Regression Actor classifier on FA-KES Titles and Contents testing dataset . . . . .	98
7.38	Classification report of Logistic Regression Actor classifier on FA-KES Titles and Contents testing dataset with tags predicted by OpenTag . . . . .	99
7.39	Classification report of the baseline LinearSVC Location classifier on FA-KES Titles testing dataset . . . . .	101
7.40	Classification report of LinearSVC Location classifier on FA-KES Titles testing dataset with tags predicted by OpenTag . . . . .	102
7.41	Classification report of the baseline Logistic Regression Location classifier on FA-KES Titles and First Paragraphs testing dataset . . . . .	103
7.42	Classification report of Logistic Regression Location classifier on FA-KES Titles and First Paragraphs testing dataset with tags predicted by OpenTag . . . . .	104
7.43	Classification report of the baseline Logistic Regression Location classifier on FA-KES Titles and Contents testing dataset . . . . .	105
7.44	Classification report of Logistic Regression Location classifier on FA-KES Titles and Contents testing dataset with tags predicted by OpenTag . . . . .	106

7.45	Classification report of the baseline LinearSVC Cause of Death classifier on FA-KES Titles testing dataset . . . . .	107
7.46	Classification report of LinearSVC Cause of Death classifier on FA-KES Titles testing dataset with tags predicted by OpenTag .	108
7.47	Classification report of the baseline LinearSVC Cause of Death classifier on FA-KES Titles and First Paragraphs testing dataset	109
7.48	Classification report of LinearSVC Cause of Death classifier on FA-KES Titles and First Paragraphs testing dataset with tags predicted by OpenTag . . . . .	109
7.49	Classification report of the baseline LinearSVC Cause of Death classifier on FA-KES Titles and Contents testing dataset . . . . .	110
7.50	Classification report of LinearSVC Cause of Death classifier on FA-KES Titles and Contents testing dataset with tags predicted by OpenTag . . . . .	111
7.51	Classification report of the baseline Logistic Regression Civilians classifier on FA-KES Titles testing dataset . . . . .	112
7.52	Classification report of Logistic Regression Civilians classifier on FA-KES Titles testing dataset with tags predicted by OpenTag .	112
7.53	Classification report of the baseline LinearSVC Civilians classifier on FA-KES Titles and First Paragraphs testing dataset . . . . .	113



7.54 Classification report of LinearSVC Civilians classifier on FA-KES  
Titles and First Paragraphs testing dataset with tags predicted by  
OpenTag . . . . . 113

7.55 Classification report of the baseline Logistic Regression Civilians  
classifier on FA-KES Titles and Contents testing dataset . . . . . 114

7.56 Classification report of Logistic Regression Civilians classifier on  
FA-KES Titles and Contents testing dataset with tags predicted  
by OpenTag . . . . . 114

# Chapter 1

## Introduction

### 1.1 Motivation

Throughout the years, technology has left a huge impact on people's lives, connecting North to South, East to West making the whole world like a spider network. In other words, every tiny information covering any topic including celebrity news, war incidents, new recipes, etc... is shared with the universe. One of the ways to stay up-to-date is social media platforms where individuals spend more than 50% of their time checking and reading the latest news. However, there are times when humans spend a lot more time on the internet and mostly when they hear about a war incident or a natural disaster that has led to casualties.

Twitter is one of the most used social media platforms where news about such incidents is tweeted sharing details about the accidents. In this project, we aim to extract information from tweets posted during the Joplin 2011 tornado and

from a set of news articles called FA-KES that report about the Syrian War. The data extracted from the articles and tweets are mainly related to the location of the incident, the number of dead humans, the damaged infrastructure, etc. For instance, if we have the following sentence “15 were dead in Aleppo”, we want to be able to extract “Aleppo” as the location of the incident and “15” as the number of dead people. This could be seen as a Named Entity Recognition (NER) task whose role is to identify and tag the name entities for a given sequence of words. We implemented a deep learning model that takes as input a sequence of tokens and retrieves specific information by tagging the words with the appropriate attribute label. The extracted information would be aggregated to perform some fact-checking that sparkle our motivation. Fact-checking consists of comparing the output of the model to ground-truth values that construct the actual value that should be detected by our system. For that to happen, we considered the answers of CrowdFlower [3] workers as ground-truth values. This will be elaborated in Chapters 4 and 5.

## 1.2 Sequence Tagging

The first step towards our goal is to label each word in the dataset with the appropriate tag. Thus, we implemented a parser that would automatically tag each word in an input sequence of tokens. This is called Sequence Tagging. Sequence Tagging can label multiple words for the same target attribute not to

mention its capability of discovering values for several attributes at the same time [1]. We will see in the remaining parts of this thesis how the advantages offered by Sequence Tagging fulfill our goal. There are many sequence tagging strategies but we used the most popular one which is the BIOES where ‘B’ stands for beginning, ‘I’ for inside, ‘O’ for outside, and ‘E’ for end of an attribute. For instance, if we had the following sequence of tokens “the incident took place in Aleppo”, and we already know that the location is Aleppo, then “Aleppo” should be tagged as ‘B-LOC’ (beginning of location) and all the other words as ‘O’. We have developed a recurrent neural network model that predicts the tags for a given sequence of words.

### 1.3 Recurrent Neural Networks

“Recurrent neural networks (RNN) capture long range dependencies between tokens in a sequence” [1] and since we’re dealing with sequential data, RNNs make the best solution. However, recurrent neural networks are known for the vanishing gradient issue which was solved by introducing Long Short Term Memory Networks [1]. Therefore, we implemented our model with the following layers: Word Embedding, Bidirectional Long Short Term Memory (BiLSTM), Attention, and Conditional Random Field (CRF). We will discuss this further in section 5.4.

## 1.4 Classifiers

For the FA-KES dataset, we were interested in identifying the actor who is behind the war incident, the location where the incident took place, the cause of death of people during the war, and the number of civilians who died during the incident. To note that we had a list of categories for actors, location, and cause of death. Therefore, we created Logistic Regression, Random Forest Classifier, and LinearSVC classifiers for each one of the mentioned attributes, picked the one that returned the highest F1-score when evaluated on the output of our deep learning model and compared its result to the result of a baseline classifier tested on the same data excluding the occurrence of tags. We will present more insight into this under section 5.5.

## 1.5 Objectives and Contributions

In this project, our goal is to retrieve values from a given sequence of tokens that correspond to the attributes that we are interested in collecting, such as the location, the number of casualties, the cause of death, etc. Several studies have been done to retrieve the answers to many questions and one of them is OpenTag [1]. We implemented the same algorithm proposed by OpenTag, trained it and tested it on Joplin and FA-KES datasets. Additionally, we created classifiers for a list of attributes seen in FA-KES articles.

Below is a summary of the contributions that have been done in this thesis:

1. A deep learning model built upon sequence tagging was developed, trained and evaluated on Joplin and FA-KES datasets.
2. A baseline model was created, trained and tested on the same Joplin and FA-KES subsets to see how well our recurrent neural network is performing.
3. Fact-checking has been done to evaluate the performance of the paradigms.
4. Classifiers were developed, trained, and examined on the output of our deep learning model.
5. Baseline classifiers were implemented, trained, and evaluated on the same FA-KES testing dataset without taking into consideration the labels.

## 1.6 Thesis Plan

The thesis is divided as follows: Chapter 2 presents the challenges that we have encountered with Joplin and FA-KES datasets, Chapter 3 gives a quick summary of previous works that have been done for information extraction, Chapter 4 gives a close look to the Joplin and FA-KES datasets, Chapter 5 describes how the majority vote approach was applied on Joplin subsets, how the answers were retrieved for the FA-KES dataset, how the sequence tagging approach was applied on both datasets, how we developed our deep learning model, how we implemented the classifiers for the FA-KES dataset, Chapter 6 gives a brief description of all the metrics that have been calculated in this study, Chapter 7

explains the evaluations of our neural network on Joplin and FA-KES subsets and the evaluations of FA-KES classifiers, and Chapter 8 ends our work with a conclusion.

# Chapter 2

## Joplin and FA-KES Challenges

Machine learning problems are built upon datasets. The more structured reliable the data is, the better the performance of deep learning systems would be. In this thesis, we encountered several challenges related to data collection, cleanup, and extraction.

To begin, we used the “Joplin” which consists of tweets collected during the tornado that struck Joplin in 2011 [4]. One of the challenges is dealing with unstructured text which would harden the extraction of target values correctly by our model. The “Joplin” subsets that we are going to see in the following sections are composed of tweets with specific information retrieved by CrowdFlower [3] workers like the location of the incident, where for each tweet, we have several answers for the same information to be collected. Therefore, we agreed on selecting one answer from the list of answers provided by the workers and this was done manually by selecting the answer that has more votes among the others.



As for the “FA-KES” dataset, it presented a lot more challenges than “Joplin”. “FA-KES” is composed of news articles collected from 15 different sources. We wanted to do experiments that cover 3 different dimensions of these articles. One of them is based on the titles and first paragraphs aggregated. However, we didn’t have the first paragraph of each article separated from its content. So, we had to collect first the URL of each article using *googlesearch* python library. After doing so, we went through the template of each of the 15 news sources to study the HTML template structure to indicate to our parser under which HTML tag the first paragraph is written. An additional challenge faced with FA-KES is having multiple HTML templates per source, thus we had to acquire them all. Additionally, more than 100 articles were collected from Al-Alam website and those articles don’t exist anymore. Therefore, we had to retrieve manually from the available content that we had the first 3-4 sentences assuming that they construct the first paragraph. Moreover, CrowdFlower [3] workers extracted values from the articles that were meant for instance to reflect the cause of death but they corresponded to the location of the event. Despite that, more than 50% of the answers didn’t exist in the articles but since we had for each information extracted the sentence from which the information was retrieved by the workers, we started to manually collect the answer from the copied sentences. Lastly, having more than 500 words in each article imposed a big challenge to our recurrent neural network model where, for instance, the location of the incident could be found in the title of the article, whereas the number of people who died at the

end of the article.

# Chapter 3

## Literature Review

We review previous works on information extraction with natural language processing and open attribute value extraction.

### 3.1 Natural Language Processing

In [4], they worked on creating an automatic system that can extract disaster-related information from tweets which would be helpful for humanitarian organizations. For the training phase, they used the “Joplin” dataset that contains tweets posted during the Joplin 2011 tornado. As for the implementation phase, two main components were required by the system: tweets classification and extraction from tweets. First, manual classification and extraction of data were done with crowdsourcing. Workers on CrowdFlower [3] had to first annotate tweets as Personal, Informative (direct, indirect, or direct/indirect), or other.

After getting the informative messages, workers classified informative tweets into one of these five categories: Caution and advice, Casualties and damages, Donations (money, goods or services), People (missing, found, or seen), and Information source. Since no tweet was classified as People, this category was removed [4]. After that, they automated tweets classification as Personal, Informative (direct, indirect, or direct/indirect), or other and information extraction using Multi-label Naïve Bayesian classifiers that were trained on the data classified by CrowdFlower [3] workers. So after filtering the informative tweets, a classifier was trained with the same features of the previous classifier and using the obtained data from crowdsourcing to identify whether an informative message is direct or indirect. Another classifier was trained to classify the informative tweets into one of the following four categories: “Caution/Advice”, “Donation”, “Casualty/Damage”, and “Information Source”. For each of the above categories, different types of information were extracted such as location references, time references, source, type, etc. and a classifier was trained for each category to automatically classify a tweet of the corresponding type into one of its sub-types. For instance, an “Advice and Caution tweet” would be classified into Warning issues or lifted, Siren heard, Shelter open or available, etc. To evaluate the quality of the information extractors, hit-ratio was measured by asking CrowdFlower [3] workers to extract manually information from a set of training tweets and extracting the same information from the same set of tweets automatically with the trained extractors. Results showed a high hit-ratio (close to 1) for “Source”

and “Number of Casualties” extractors. However, the performance of the other extractors was poor which enlighten the necessity of using more sophisticated algorithms for the extraction process [4].

Their future work [5] was focused on developing more sophisticated extractors that use complex Natural Language Processing techniques. They worked on extracting disaster-relevant information from tweets with NLP. The first step was collecting tweets that are relevant to disasters. “Joplin 2011” and “Sandy 2012” datasets were collected through Twitter’s API using hashtags “#joplin” and “#sandy” respectively. The next step was classifying tweets, whether a tweet is personal or informative. Once an informative tweet is detected, they classify it as one of the following classes: “caution and advice”, “casualties and damage”, “donations”, “people”, “information sources”, “other” using the multi-label classifiers from their previous work [4]. And now after classifying tweets, they were able to extract relevant information. So, they used the conditional random field (CRF) which is a probabilistic model that predicts the relevant words to the disaster in a tweet, marks them with a ‘+’ symbol and labels the remaining with a ‘-’ symbol. They used Arknlp which is “an implementation of CRFs and a set of features known to be effective for NLP tasks on Twitter” [5]. During the extraction phase, the workers were given the tweet with its type (caution or advice, people, etc.), its instruction (a description of the type), and an empty text field where they had to copy/paste words from the corresponding tweet that verify the given instruction. Next, they trained their system on a part of the human-provided labels

and tested it on the remaining part. Two metrics were taken into consideration: detection rate (fraction of examples in which humans and system found something relevant besides if it's correct or not), hit ratio (fraction of examples in which system found something relevant which is considered correct by humans) [5]. These two metrics were measured to evaluate the performance of the information extraction phase for several configurations of training and testing set on both datasets, Joplin and Sandy. The experiments took place on the largest classes detected in those two datasets: “caution and advice”, “casualties and damage”, “infrastructure”, and “donations”. It was observed that a high hit ratio is usually associated with a lower detection rate and vice versa. Results showed that the system was able to detect relevant information from 40% to 80% (detection rate) and generate correct output 80% to 90% (hit ratio) of the time which means that this developed system can extract reliable high-level information [5].

## 3.2 Open Attribute Value Extraction

A deep learning model named OpenTag was introduced in [1] and applied on product profiles retrieved from *Amazon.com* public pages. The goal of this model is to extract values for a set of pre-defined target attributes that might be missing from the list of a product's attributes. OpenTag maintains the Sequence Tagging approach which by itself models the dependency between attribute values. Thus, OpenTag could be categorized as NER (Named Entity Recognition) task that

aims to tag each word in a given sequence of tokens using the Sequence Tagging approach. There are many sequence tagging strategies, and the most popular one is used in this study which is BIOES-style where ‘B’ stands for the beginning of an attribute, ‘I’ stands for inside of an attribute, ‘O’ stands for outside of an attribute and ‘E’ stands for end of an attribute. Sequence tagging allows the extraction of multi-word attribute values, the extraction of multiple attribute values at the same time, and tagging all tokens whether they are attribute values or not. The main advantages of Sequence tagging consist of: discovering values for several attributes at the same time, labeling multiple words for the same attribute [1].

OpenTag doesn’t rely on a dictionary of words. This model consists of the following layers: word embedding, Bi-LSTM, attention, and CRF. Word embeddings generate a vector for each token. The output of the first layer would be passed as input to a bidirectional long short-term memory (Bi-LSTM) layer. Attention mechanism is applied to the output of Bi-LSTM. To ensure coherency between attribute tags during prediction, Conditional Random Fields is used as the last layer of the OpenTag model. OpenTag was implemented using TensorFlow and experiments were performed on three types of product profiles: dog food, detergents, and cameras. They compared Bi-LSTM, Bi-LSTM-CRF and OpenTag models’ results on the three product domains. OpenTag outperformed all models on all different datasets used for single and multiple attributes values extraction with an overall 82.8% F-score. This end-to-end tagging model could be applied to any text with high performance [1].

# Chapter 4

## Dataset

### 4.1 Joplin Dataset

In our project, we will use the “Joplin” dataset which consists of tweets posted during the Joplin 2011 tornado. We are going to tackle four subsets of the Joplin dataset: Caution and Advice, Casualties and Damages, Donations, and Information Source. These subsets are taken from previous works [4, 5] where CrowdFlower [3] workers had to extract specific information from a list of tweets. In some cases, workers wrote “N/A” or “n/a” which both stand for “not available”.

Casualties and Damages dataset is composed of 138 tweets. Each tweet in this dataset presents information about losses caused by an incident. Crowdsourcing workers had to extract from a set of tweets the number of people who died or were injured during the incident, and the infrastructure that was damaged. While checking the answers of the workers, we could see that more than one answer was



provided per tweet as can be seen in Table 4.1. For instance, in the first tweet, each worker responded to the question of ‘How many injured or dead people’ as either “house”, or “n/a”.

Tweet	How many injured or dead people	Damaged Infrastructure
My house was blown away by a tornado, but a famous person just‘d something about my town. Who needs a house?	house house n/a n/a n/a	house house n/a n/a n/a house
@BBCWorld: US authorities confirm at least 89 killed after massive tornado hits Joplin, southwest Missouri	89 89 89	n/a
Tornado is gone and woman is assessing the damage to her house and still can barely stand because of the wind.	n/a	house n/a house

Table 4.1: Sample of Casualties and Damages’ tweets

The Caution and Advice dataset consists of 438 tweets that warn or advise about an incident that may happen. The workers had to extract information about the incident itself, the location where the incident took place, and the time of the incident. In this case, also, multiple answers were provided for each question as can be seen in Table 4.2. For example, the location for the first tweet was filled as either “CT”, or “N/A”.

Tweet	Incident	Location	Time
@Juss2Live: News was goin crazy about tornado warnings n it didn even rain ! -_- - - shit was wild over in CT . 4 ppl died	tornado warnings tornado warnings tornado warnings	CT N/A CT	n/a
@spann: Everybody on the campus of the University of Oklahoma should be in a tornado safe place now. #okwx	should be in a tornado safe place — should be in a tornado safe plac — tornado safe place	Oklahoma — campus of the University of Oklahoma — campus of the University of Oklahoma	
Almyra Arkansas tornado about to hit the ground	tornado — tornado about to hit the ground — tornado about to hit the ground	Almyra Arkansas Almyra Arkansas Almyra Arkansas	

Table 4.2: Sample of Caution and Advice tweets

As for the Donations dataset, it consists of 204 tweets where each tweet speaks about fundraising, donations offered or asked by the victims of an incident. Workers retrieved from the tweets the donation offer, the location of the donation and the authority responsible of this donation or asking for a donation. In Table 4.3,

several answers for the donation offer were extracted for the third tweet: “water”, “collect water”, “water”.

Tweet	Donation	Location	Authority
@caryrandolph: It’s official: Margaritas for #Joplin next Wednesday	100% of proceeds —	Ludlow St.	@losFeliz_NYC on Ludlow St.
@losFeliz_NYC on Ludlow St. 100% of proceeds go to OzarksRed- Cross.	100% of proceeds go to @OzarksRedCross.	Ludlow St.	— @losFeliz_NYC
@SLMPD: 60 of our officers are on their way to Joplin, Mo to assist as the city recovers from the May 22 tornado. <a href="http://twitpic.com">http://twitpic.com</a> ...	assist assist	Joplin, Mo Joplin, Mo	60 of our officers — officers
@GabeCarimi: At West Town Mall helping out 93.1 Jamz collect water for the tornado victims in Joplin. They’re here till 6, stop by a ...	water collect water water	West Town Mall — At West Town Mall — West Town Mall	93.1 Jamz 93.1 Jamz 93.1 Jamz

Table 4.3: Sample of Donations’ tweets

Lastly, the Information Source dataset contains 280 tweets where each tweet

carries a photo or video related to an incident. The workers' job was to take out of the tweet the incident that happened and its information source. Table 4.4 shows two different answers for the incident concerning the second tweet: "Tornado", "Tornado sucks up a river".

## 4.2 FA-KES Dataset

In our study, we will also use "a fake news dataset around the Syrian war", called FA-KES which is presented in [6]. This dataset consists of news articles that announce Syrian war incidents. FA-KES is composed of 804 English articles that were retrieved from "several media outlets representing mobilisation press, loyalist press, and diverse print media" [6].

In Table 4.5, we present a sample of FA-KES' articles for which CrowdFlower [3] workers had to answer the list of questions below after reading its content:

1. How many civilians died in the incident?
2. How many children were targeted in the incident?
3. How many adult women were targeted in the incident ?
4. How many non-civilians died in the incident?
5. What is the cause of death?
6. Who does the article blame for the casualties?

Tweet	Incident	Information Source
@lifechurchtv: See an update of how LifeChurch.tv is partnering with Tornado Relief in Joplin, Oklahoma City and Birmingham. <a href="http://y...">http://y ...</a>	@lifechurchtv: See an update of how LifeChurch.tv is partnering with Tornado Relief in Joplin, Oklahoma City and Birmingham. — @lifechurchtv: See an update of how LifeChurch.tv is partnering with Tornado Relief in Joplin, Oklahoma City and Birmingham. — See a	LifeChurch.tv — <a href="http://y">http://y</a> — <a href="http://y/">http://y/</a> — <a href="http://y .">http://y .</a>
Tornado sucks up a river. <a href="http://is.gd/tutjk0">http://is.gd/tutjk0</a>	Tornado — Tornado sucks up a river	<a href="http://www.bbc.co.uk/news/world-us-canada-13627044">http://www.bbc.co.uk/news/world-us-canada-13627044</a> — <a href="http://is.gd/tutjk0">http://is.gd/tutjk0</a>
Joliet councilman organizes tornado relief collection - Herald News <a href="http://t.co/PqMTvXw">http://t.co/PqMTvXw</a> via @AddThis	Joliet councilman organizes tornado relief collection - Herald News — tornado relief collection	<a href="http://t.co/PqMTvXw">http://t.co/PqMTvXw</a> — <a href="http://t.co/PqMTvXw">http://t.co/PqMTvXw</a>

Table 4.4: Sample of Information Source tweets

7. Where does the article claim the deaths happened?
8. When did the incident happen (Day/Month/Year)?

To begin with our experiments, we created two subsets of the news articles dataset. The first one consists of the articles' titles with the workers' answers and the second one of the articles' titles concatenated with their contents accompanied by the workers' answers. At some point, we wanted to extend our experiments to the first paragraph of each article; however, we didn't have the first paragraph separately and for many articles, we didn't have the URL from which the article was originally collected. Thus, we worked on extracting the URL of each article using *googlesearch* python library which takes as input the article title and source. One thing to note here that we have 15 different article sources. After collecting the hyperlinks, we started learning the structure of the HTML template of each web source so we could tell our system where the first paragraph is located, under which HTML tag specifically, to be able to retrieve it. Also, more than one HTML template was used per news source which hardened our work and made us go through all the available templates. Plus, while checking the articles online, we couldn't find the ones retrieved from the Al-Alam website (around 200 articles). Thus, from the available content that we had initially, we retrieved the first 3-4 sentences which we assumed would construct the first paragraph. In this case, the third subset created is composed of the articles' titles concatenated with the first paragraphs along with the workers' answers.

Article	Air raids kill 11 civilians in east Syria Monitor AFP Friday 15 Jul 2016 "At least 11 civilians – among them four women and four children – were killed in Syrian or Russian air raids on the Al- Boulil region that is controlled by the Islamic State group in the eastern province of Deir Ezzor" the Britain-based monitor said. ...
Cause of death	warplane shelling
Actor	syrian government and affiliated militias
Number of dead civilians	11
Number of dead non- civilians	0
Number of dead children	4
Number of dead women	4
Place of death	deir Ezzor
Date of incident	7/15/2016

Table 4.5: Sample of FA-KES articles

# Chapter 5

## Proposed Approach

### 5.1 Joplin Dataset

#### 5.1.1 Majority Vote

As seen in Tables 4.1, 4.2, 4.3, and 4.4, multiple answers were given by the workers for each question which may cause inconsistency in our dataset. To avoid it, we selected only one answer from the list of answers provided.

The selection of responses is achieved by the concept of “Majority Vote”. The answer that has more occurrences would be selected and in case there are different answers or the number of votes for each suggested answer are equal, we would have to select the most suitable answer.

In Table 4.1, we could see for the first tweet, that five answers are present for question ‘How many injured or dead people’: three answers ‘n/a’ vs two answers



‘house’, so in this case ‘n/a’ wins and the answer selected is ‘n/a’. However, for the second question, there are six answers for “Damaged Infrastructure”: three answers ‘n/a’ vs three answers ‘house’. In such cases, we had to select the most convenient answer which would be here ‘house’ of course.

## 5.2 FA-KES Dataset

### 5.2.1 Answer Retrieval

As seen in Table 4.5, the information extracted by the workers for attribute actor, for example, doesn’t exist as it is in the article’s content because CrowdFlower [3] workers were given a list of values to pick from for each question restricting their choice. Also, these lists were actually given by the Syrian Violations Documentation Center. What was helpful in this case, is having for each information extracted by the workers, the sentence that reflects their answer.

Therefore, we started to retrieve manually the attributes’ values from the copied sentences. For instance, in Table 4.5, workers picked “syrian government and affiliated militias” as actor for that particular article and this series of tokens was not found in the text. With the help of the sentence copied from which this information was retrieved “At least 11 civilians – among them four women and four children – were killed in Syrian or Russian air raids on the Al-Boulil region that is controlled by the Islamic State group in the eastern province of Deir

Ezzor” [7], we retrieved the authority responsible of the incident which should be “Syrian or Russian” in this case as is shown in Table 5.1. Besides, we noticed that the value for the date attribute was extracted by the workers in the following format *month/day/year*. However, each article contains the date in a specific format. While checking the articles, we were able to identify 15 different date representation. Therefore, we worked on converting the dates retrieved by the workers to those 15 different representations which will help the parser in the sequence tagging step to automatically check if one of those formats exists in the article to be able to tag them with the appropriate date labels. For instance, one of the formats of the date retrieved in Table 4.5 is **Friday 15 Jul 2016** which actually exists in the article.

### 5.3 Sequence Tagging

After getting fully cleaned data, we used the BIOES Sequence Tagging’s strategy which was utilized in OpenTag [1] in the aim of getting every word in the dataset associated to a label called ‘tag’. A tag consists of one of these letters B, I, O, or E, that stand respectively for beginning, inside, outside, or end of an attribute, followed by a ‘-’ sign, followed by three letters that represent the type of information that was initially extracted by the CrowdFlower [3] workers. In other words, if the information extracted for the location of the incident from a particular tweet or article is for example ‘inside the pizza shop’, the first word

Attribute	Workers' Answer	Copied From	Final Answer
Cause of death	warplane shelling	At least 11 civilians – among them four women and four children – were killed in Syrian or Russian air raids on the Al-Boulil	air raids
Actor	syrian government and affiliated militias	At least 11 civilians – among them four women and four children – were killed in Syrian or Russian air raids on the Al-Boulil	Syrian or Russian
Number of dead civilians	11	At least 11 civilians – among them four women and four children – were killed in Syrian or Russian air raids on the Al-Boulil	11 civilians
Number of dead non-civilians	0		
Number of dead children	4	At least 11 civilians – among them four women and four children – were killed in Syrian or Russian air raids on the Al-Boulil	four children
Number of dead women	4	At least 11 civilians – among them four women and four children – were killed in Syrian or Russian air raids on the Al-Boulil	four women
Place of death	deir Ezzor	controlled by the Islamic State group in the eastern province of Deir Ezzor	Deir Ezzor
Date of incident	7/15/2016		Friday 15 Jul 2016

Table 5.1: Answers extracted from FA-KES' article shown in Table 4.5

of the location which is here ‘inside’, would be tagged as beginning of location ‘B-LOC’, the last word ‘shop’ would be labeled as ‘E-LOC’ which stands for end of location, and any word in between these two words would be marked as ‘I-LOC’ which stands for inside of location; in this example, ‘the’, and ‘pizza’, are both tagged as ‘I-LOC’. If the remaining words of the tweets or articles were not retrieved by the workers, each token would be tagged as ‘O’ which designates outside of an attribute (words that are outside the scope of information that we are interested in). To automate this work, we implemented a parser for the four subsets of the Joplin dataset and the three subsets of the FA-KES dataset.

For the Casualties and Damages dataset, our parser would take as input the list of tweets with the information extracted for every question, in this case, the number of people who were dead or injured and the infrastructure that was damaged during the incident. According to BIOES-style strategy, we would have the following labels: ‘B-PEO’, ‘I-PEO’, ‘E-PEO’, ‘B-INF’, ‘I-INF’, ‘E-INF’, and ‘O’ (where O stands for words outside the scope, PEO stands for the injured or dead people, and INF for the damaged infrastructure). In Table 5.2, we present the sequence of words of the third tweet shown in Table 4.1 with their appropriate tags.

Word	Tag
Tornado	O
is	O
gone	O
and	O
woman	O
is	O
assessing	O
the	O
damage	O
to	O
her	O
house	B-INF
and	O
still	O
can	O
barely	O
stand	O
because	O
of	O
the	O
wind.	O

Table 5.2: Example of Casualties and Damages’ tweet

BIOE was applied to the three other Joplin datasets. Words in Caution and Advice tweets ended up tagged with the following tags: ‘B-INC’, ‘I-INC’, ‘E-INC’, ‘B-LOC’, ‘I-LOC’, ‘E-LOC’, ‘B-TIM’, ‘I-TIM’, ‘E-TIM’, and ‘O’ (where O stands for words outside the scope, INC for incident, LOC for location of incident, TIM for time of incident). Table 5.3 shows the tags for the series of terms that occurred in the third tweet of Table 4.2.

Word	Tag
Almyra	B-LOC
Arkansas	E-LOC
tornado	B-INC
about	I-INC
to	I-INC
hit	I-INC
the	I-INC
ground	E-INC

Table 5.3: Example of Caution and Advice tweet

As for the Donations' tweets, the series of words were marked as either 'B-DON', 'I-DON', 'E-DON', 'B-LOC', 'I-LOC', 'E-LOC', 'B-REP', 'I-REP', 'E-REP', or 'O' (where 'O' stands for words outside the scope, 'DON' for the donation event, 'LOC' for the location of the donation event, and 'REP' for the authority responsible of the donation event or asking for a donation). Table 5.4 presents the output of the parser for the second tweet presented in Table 4.3.

Word	Tag
@SLMPD:	O
60	B-REP
of	I-REP
our	I-REP
officers	E-REP
are	O
on	O
their	O
way	O
to	O
Joplin,	B-LOC
Mo	E-LOC
to	O
assist	B-DON
as	O
the	O
city	O
recovers	O
from	O
the	O
May	O
22	O
tornado.	O
<a href="http://twitpic.com">http://twitpic.com</a>	O
...	O

Table 5.4: Example of Donations’ tweet

Concerning Information Source tweets, each token was tagged with one of these labels: ‘B-INC’, ‘I-INC’, ‘E-INC’, ‘B-SRC’, ‘I-SRC’, ‘E-SRC’, or ‘O’ (where O stands for words outside the scope, INC for incident, and SRC for information source). In Table 5.5, we present the labels for each word in the first tweet of Table 4.4.

Word	Tag
@lifechurchtv:	B-INC
See	I-INC
an	I-INC
update	I-INC
of	I-INC
how	I-INC
LifeChurch.tv	I-INC
is	I-INC
partnering	I-INC
with	I-INC
Tornado	I-INC
Relief	I-INC
in	I-INC
Joplin,	I-INC
Oklahoma	I-INC
City	I-INC
and	I-INC
Birmingham.	E-INC
http://y	B-SRC
...	O

Table 5.5: Example of Information Source tweet

Finally, tokens in FA-KES’ articles were labeled with one of the following tags: ‘B-LOC’, ‘I-LOC’, ‘E-LOC’, ‘B-CIV’, ‘I-CIV’, ‘E-CIV’, ‘B-NCV’, ‘I-NCV’, ‘E-NCV’, ‘B-WMN’, ‘I-WMN’, ‘E-WMN’, ‘B-CHD’, ‘I-CHD’, ‘E-CHD’, ‘B-ACT’, ‘I-ACT’, ‘E-ACT’, ‘B-COD’, ‘I-COD’, ‘E-COD’, ‘B-DAT’, ‘I-DAT’, ‘E-DAT’, or ‘O’ (where O stands for words outside the scope, LOC for location of incident, CIV for number of civilians dead, NCV for number of non-civilians dead, WMN for number of women targeted, CHD for number of children killed, ACT for actor/authority responsible of incident, COD for cause of death, and DAT for date of incident). Table 5.6 shows the appropriate tag for each word in the ‘Final



Answer' column seen in Table 5.1.

Word	Tag
Friday	B-DAT
15	I-DAT
Jul	I-DAT
2016	E-DAT
11	B-CIV
civilians	E-CIV
four	B-WMN
women	E-WMN
four	B-CHD
children	E-CHD
Syrian	B-ACT
or	I-ACT
Russian	E-ACT
air	B-COD
raids	E-COD
Deir	B-LOC
Ezzor	E-LOC

Table 5.6: Example of FA-KES' articles

## 5.4 Deep Learning Model

Our proposed model is inspired by the OpenTag [1] approach. We used Word Embedding as a first layer, where each token is represented by an integer number leading to having the whole sentence represented by a vector of integer numbers. The output of the embedding layer represents the input to our Bi-LSTM layer.

Long Short Term Memory Networks (LSTM) is a sub-category of Recurrent Neural Networks (RNN). LSTM cell's job consists of generating a hidden vector  $h_t$  for each token  $x_t$  represented by its embedding  $e_t$  which is passed as input

to the LSTM cell. That way the generated vector would be passed as input to the next layer. In our approach, since we are using sequence tagging which by itself needs to look up to previous and future contexts, we considered applying a Bidirectional-LSTM instead of LSTM where now we have two hidden vectors one for backward, and one for forward which are concatenated to form the final output as a new hidden vector  $h_t$  [1]. From [1], we will recall the representation of  $h_t$ :

$$h_t = \sigma([\vec{h}_t, \overleftarrow{h}_t]) \quad (5.1)$$

However, BiLSTM is not enough for our sequence tagging approach. BiLSTM lacks in measuring the coherency of tags given as input a sequence of words. For instance, if we have the following sequence of words “Starbucks coffee shop”, we might get as labels ‘B-LOC’, ‘E-LOC’, and ‘I-LOC’, respectively; whereas, the ‘E-LOC’ shouldn’t appear before ‘I-LOC’. This could only be fixed by adding a Conditional Random Field (CRF) layer which focuses on predicting the label sequence jointly. In this example, CRF would predict ‘B-LOC I-LOC E-LOC’ for “Starbucks coffee shop”. As is shown in [1], CRF function can be written as follows:

$$Pr(\mathbf{y}|\mathbf{x}; \Psi) \propto \prod_{t=1}^T \exp\left(\sum_{k=1}^K \Psi_k f_k(y_{t-1}, y_t, x)\right) \quad (5.2)$$

where “ $x = \{x_1, x_2, \dots, x_n\}$  is the input sequence,  $y = \{y_1, y_2, \dots, y_n\}$  is the corresponding label sequence,  $f_k(y, x)$  is the feature function,  $\Psi_k$  is the corresponding weight to be learned,  $K$  is the number of features,  $y_t$  and  $y_{t-1}$  are the neighboring tags at timesteps  $t$  and  $t-1$ , respectively” [1]. We also added an attention layer on top of our model which would “highlight important concepts, rather than focusing on all the information” [1]. Thus, we ended up having the same algorithm as OpenTag. “The attention-focused hidden state representation  $l_t$  of a token at timestep  $t$  is given by the weighted summation of the hidden state representation  $h_{t'}$  of all other tokens at timesteps  $t'$ , and their similarity  $\alpha_{t,t'}$  to the hidden state representation  $h_t$  of the current token” [1]:

$$l_t = \sum_{t'=1}^n \alpha_{t,t'} \cdot h_{t'} \quad (5.3)$$

To avoid over-fitting, we applied the L2 regularization technique to our BiLSTM layer using the regularizers from Keras [8]. We trained the model on 80% of the data available in each dataset and tested on the remaining 20%. Besides, we validated the paradigm on 20% of the training dataset. We will see in Chapter 7 how well our model performed on each of the datasets.

## 5.5 FA-KES Classifiers

In [6], CrowdFlower [3] workers were given a set of categories to pick from the answers to a list of questions. These categories were provided by the Syrian

Violations Documentation Center for the following questions: ‘Who does the article blame for the casualties?’, ‘What is the cause of death?’ and ‘Where does the article claim the deaths happened?’. So, now instead of looking at an open-tag environment, we are looking at a closed-tag one since we already know the list of answers for each one of the above-mentioned questions. We wanted to see how well would a closed-tag problem perform on the output of our OpenTag model. Therefore, we created Logistic Regression, Random Forest Classifier, and LinearSVC classifiers for each of the following attributes: Actor, Location, Cause of Death, and Civilians. “Logistic regression analysis is one of the mostly preferred regression methods that can be implemented in modelling binary dependent variables. Logistic regression is a mathematical modelling approach used to define the relationship between such independent variables as  $X_1, X_2, \dots, X_n$  and  $Y$  binary dependent variable which is coded as 0 or 1 for two possible categories. The independent variables may be continuous, discrete, binary or a combination of them.” [9]. We applied the One-vs-Rest approach for the Logistic Regression classifier which “takes one class as positive and rest all as negative and trains the classifier. So for the data having  $n$ -classes it trains  $n$  classifiers. Now in the classification phase the  $n$ -classifier predicts probability of particular class and class with highest probability is selected.” [10]. “Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.” [11]. The last classifier is called LinearSVC. “The objective of a

Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a “best fit” hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the “predicted” class is. This makes this specific algorithm rather suitable for our uses, though you can use this for many situations.” [12]. These classifiers were evaluated on the output of our deep learning model.

### **5.5.1 Actor Classifier**

CrowdFlower [3] workers had to select from the following list of actors the actor who is behind the incident: Al-Nusra Front, Armed opposition groups, International coalition forces, Russian troops, Self administration forces, Syrian government and affiliated militias, The organization of Islamic State in Iraq and the Levant - ISIS, and Other. We noticed that the training dataset is imbalanced, so we weighted it by giving the least occurring classes a higher example weight. Then, we created a Logistic Regression, Random Forest Classifier classifiers, and LinearSVC classifiers for actor classification that were trained respectively on the three weighted subsets: the articles’ titles and their appropriate ground-truth labels, the articles’ titles concatenated with first paragraphs and their corresponding tags, and the articles’ titles concatenated with their contents and their relative labels. These classifiers were then tested on the output of our deep learning model for the 3 dimensions: titles, titles and first paragraphs, titles and

contents. To see how well our classifiers are doing, the same classifiers were created and trained on the same weighted datasets excluding the labels. Finally, we examined these basic classifiers on the same testing dataset without taking into consideration the occurrence of tags.

### 5.5.2 Location Classifier

In [6], workers had to pick the location of the incident from the following list: Aleppo, Damas, Damascus, Damascus Suburbs, Daraa, Deir Ezzor, Hama, Hasakeh, Homs, Idlib, Lattakia, Quneitra, Raqqa, Tartous, and Other. In this case, we noticed that the training datasets are not balanced. Thus, we weighted all the occurring classes in the training dataset by giving a higher weight to the least occurring classes. Logistic Regression, Random Forest Classifier, and LinearSVC classifiers were created for location classification and trained respectively on the weighted datasets: the articles' titles and their appropriate ground-truth labels, the articles' titles concatenated with first paragraphs and their corresponding tags, and the articles' titles concatenated with their contents and their relative tags. We evaluated the performance of those classifiers on the output of our deep learning model for the 3 dimensions: titles, titles and first paragraphs, titles and contents. Baseline classifiers were implemented and trained on the same weighted subsets disregarding the existence of labels. Lastly, we tested the baseline classifiers on the testing data excluding labels.

### 5.5.3 Cause of Death Classifier

The following list was presented to the workers to choose from the cause of death: Chemical and toxic gases, Execution, Explosion, Kidnapping - Execution, Kidnapping - Torture, Shelling, Shooting, Warplane shelling, and Other. We could see that there are some repetitive categories, so we grouped “Execution” and “Kidnapping - Execution” as “Execution”, and “Warplane shelling” and “Shelling” as “Shelling”. We also removed the data that has “Kidnapping - Torture” as a cause of death category from the training dataset. After that, we weighted our imbalanced training dataset the same way we did it for the Location and Actor classifiers. Logistic Regression, Random Forest Classifier, and LinearSVC classifiers were created for cause of death classification and trained respectively on the weighted datasets: the articles’ titles and their appropriate ground-truth labels, the articles’ titles concatenated with the first paragraphs and their corresponding tags, and the articles’ titles concatenated with their contents and their relative tags. We evaluated the performance of those classifiers on the output of our deep learning model for the three dimensions: titles, titles and first paragraphs, titles and contents. Besides that, baseline classifiers were implemented and trained on the same training datasets disregarding the existence of any labels. At last, we examined the performance of those baseline classifiers on the similarly testing subsets without tags.

#### 5.5.4 Civilians Classifier

As mentioned earlier, we had the copied sentence from which the workers extracted the number of dead civilians. We managed to round the digit numbers extracted as value for the number of dead civilians for the three dimensions that we are tackling: the articles' titles, the articles' titles concatenated with first paragraphs, and the articles' titles concatenated with their contents. After rounding these values, we linked them to one of these groups that represent the number of civilians who were dead during the war incident: 'Less than 50', 'Greater than 50', or 'Greater than 100'. We noticed that also in this case the dataset is imbalanced. Therefore, we weighted all the occurring classes in the training dataset by giving more weight to the least occurring classes. Logistic Regression and LinearSVC classifiers were created for dead civilians classification and trained respectively on the weighted datasets: the articles' titles and their appropriate ground-truth labels, the articles' titles concatenated with first paragraphs and their corresponding tags, and the articles' titles concatenated with their contents and their relative tags. We evaluated the performance of those classifiers on the output of our deep learning model for the three dimensions: titles, titles and first paragraphs, titles and contents. Baseline classifiers were implemented and trained on the likewise weighted subsets neglecting the presence of any labels. Lastly, we evaluated the baseline classifiers on the text solely of the testing subsets.



# Chapter 6

## Metrics

In this chapter, we will present a set of metrics that we used to evaluate our model. The range for these metrics is  $[0, 1]$ . Some of the metrics shown in this chapter, use for their computation True Positive, True Negative, False Positive, and False Negative, which are defined by [2] in Table 6.1. Our focus is mainly on the values obtained for F1-score and Bleu-score.

The first metric is called *Recall*, which represents “the proportion of positive data points that are correctly considered as positive, with respect to all positive data points” [13]. The following formula illustrates the calculation of *Recall* which is proved by [13]:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (6.1)$$

“While recall expresses the ability to find all relevant instances in a dataset, precision expresses the proportion of the data points our model says was relevant actually were relevant.” [2]. From [2], *Precision* could be calculated as follows:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (6.2)$$

It is known that better performance is usually linked to higher F1-score. F1-score constitutes “the harmonic mean of precision and recall” [2], which makes them essential for its calculation, as can be seen in the below equation taken from [2]:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (6.3)$$

A lot of studies use *Accuracy* as an essential metric to evaluate their models because it measures the proportion of the number of correct predictions to the total number of predictions made, and which is generally the main goal of new proposed approaches. The below equation is taken from [14]:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (6.4)$$

In [5], they defined *Hit-ratio* as “Hit-ratio measures the fraction of examples for which our system found something, and that something could be considered correct by humans. We consider the output correct if it overlaps in at least one word with the given human label.”. The below formula is taken from [4], which is a previous work of [5], where *hit* would be equal to 1 if one token per tweet gets labeled correctly:

$$Hit - ratio = \frac{\sum_{i=1}^{|tweets|} hit_i}{|tweets|} \quad (6.5)$$

Detection rate was also defined in [5] as “Detection rate measures the fraction of examples in which humans found a relevant piece of information, and our system also found something, even if that something is incorrect.”. Since there’s no mathematical formula from previous works that present how it was computed, we wrote equation 3.6 that shows how the Detection-rate is actually measured, where *detect* would be equal to 1, whether a word gets predicted correctly or not.

$$Detection - rate = \frac{\sum_{i=1}^{|tweets|} detect_i}{|tweets|} \quad (6.6)$$

Finally, the Bilingual Evaluation Understudy Score, known as BLEU, was proposed in [15] for evaluating a generated sentence to a reference sentence. Having candidate translations and reference sentences, the approach counts the matching n-grams between the candidate sentence and the reference sentences provided, where for instance a bi-gram consists of comparing every pair of words. We measured it using *sentence\_bleu()* function provided by *NLTK* [16]. In the following lines, we present the computation of BLEU proved by [15]:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \quad (6.7)$$

$$BLEU = BP \cdot \exp \left( \sum_{n=1}^N w_n \log p_n \right) \quad (6.8)$$

where “*BP* is the brevity penalty, *c* the length of the candidate translation, *r* the effective reference corpus, *N* length of the grams, *p<sub>n</sub>* n-gram precision, and *w<sub>n</sub>* positive weights” [15]. In our study, we used the default 4-grams BLEU score.

True Positive	data points labeled as positive that are actually positive
False Positive	data points labeled as positive that are actually negative
True Negative	data points labeled as negative that are actually negative
False Negative	data points labeled as negative that are actually positive

Table 6.1: “Four Outcomes of Binary Classification” taken from [2]

# Chapter 7

## Evaluations

### 7.1 Evaluation of Deep Learning on Joplin Dataset

To recognize how well our deep learning model is doing, we created a baseline model to be examined on the same testing data. One thing to note here that our OpenTag model didn't return good results when trained on the Joplin dataset; however, when we removed the attention layer, it produced better results. Thus, the deep learning model that is trained and tested on the Joplin subsets is a BiLSTM-CRF. The baseline model consists of the same layers of our BiLSTM-CRF model excluding the Bi-LSTM layer, so we ended up having a baseline model composed of a Word Embedding layer followed by a CRF layer. We agreed on calling the baseline model CRF.

As we know, there are four subsets in the Joplin dataset. Accordingly, we created a BiLSTM-CRF model and CRF model for each of these subsets. Both

models were trained on the respective training data and tested on the corresponding data of each subset. All of this was done to compare the results of our deep learning model to the ones obtained by the baseline model and see if our deep learning model will outperform the CRF model. We trained the models for 200 epochs with a batch of size 32 and used RMSProp as an optimizer. We measured the metrics listed in Chapter 6 for the CRF and BiLSTM-CRF models and compared the values. Also, we drew the learning curve of the loss of the training and validation datasets for each of the evaluated models.

### **7.1.1 Casualties and Damages Dataset**

We split the Casualties and Damages dataset into 60% for training, 20% for validating, and 20% for testing the model. We trained the CRF and BiLSTM-CRF models on the learning data and tested them on the testing dataset. Table 7.1 shows the values of the calculated metrics for both models. Tables 7.2 and 7.3 present the classification report of our evaluated models CRF and BiLSTM-CRF, respectively. Figures 7.1 and 7.2 show the learning curve of the loss of both models on the training and validation datasets.

<b>Metric</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
F1-score	0.625	0.40
Accuracy	0.988	0.985
Hit-ratio	0.178	0.107
Detection Rate	0.178	0.143
Bleu-Score	0.977	0.971

Table 7.1: Results of CRF and BiLSTM-CRF on Casualties and Damages testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
PEO	1	0.40	0.57	10
INF	1	1	1	1
Average/Total	1	0.45	0.61	11

Table 7.2: Classification report of CRF model on Casualties and Damages testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
PEO	0.75	0.3	0.43	10
INF	0	0	0	1
Average/Total	0.68	0.27	0.39	11

Table 7.3: Classification report of BiLSTM-CRF on Casualties and Damages testing dataset



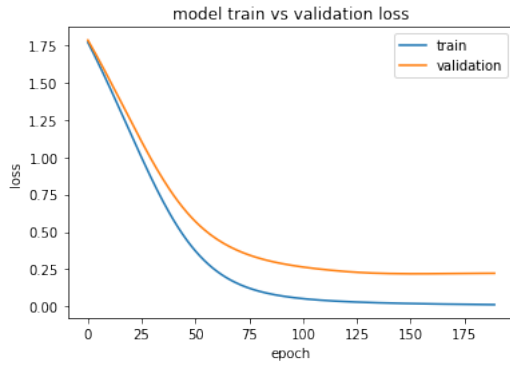


Figure 7.1: CRF loss on Casualties and Damages Training and Validation Datasets

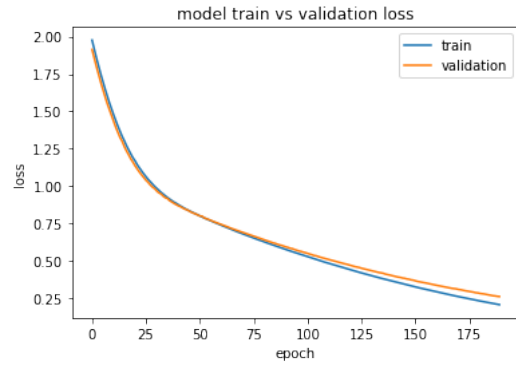


Figure 7.2: BiLSTM-CRF loss on Casualties and Damages Training and Validation Datasets

### 7.1.2 Caution and Advice Dataset

Caution and Advice dataset was also divided into 60% for training, 20% for validating, and 20% for testing the model. CRF and BiLSTM-CRF were trained on the training dataset and evaluated on the testing dataset. Table 7.4 shows the metrics values for both models. Tables 7.5 and 7.6 present the classification report of our tested models CRF and BiLSTM-CRF, respectively. The loss of both models on the training and validation datasets can be seen in Figures 7.3 and 7.4.

<b>Metric</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
F1-score	0.49	0.537
Accuracy	0.887	0.895
Hit-ratio	0.91	0.863
Detection Rate	0.977	0.966
Bleu-Score	0.851	0.873

Table 7.4: Results of CRF and BiLSTM-CRF on Caution and Advice testing dataset

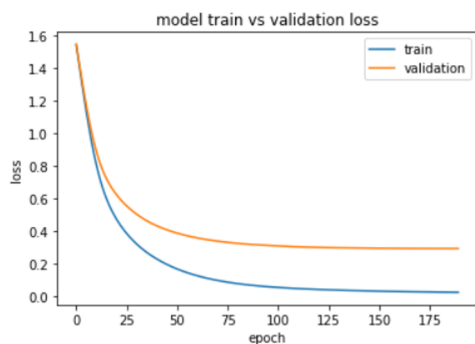


Figure 7.3: CRF loss on Caution and Advice Training and Validation Datasets

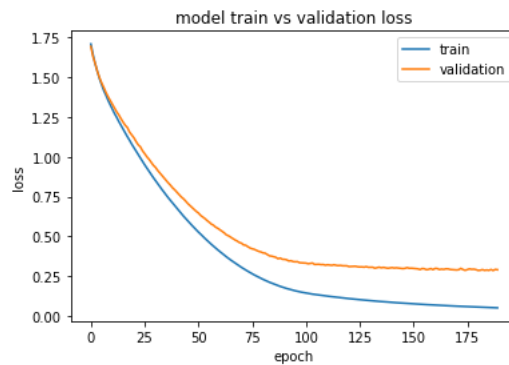


Figure 7.4: BiLSTM-CRF loss on Caution and Advice Training and Validation Datasets

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
INC	0.63	0.69	0.66	85
LOC	0.31	0.28	0.29	58
TIM	0.31	0.33	0.32	27
Average/Total	0.47	0.49	0.48	170

Table 7.5: Classification report of the CRF model on Caution and Advice testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
INC	0.66	0.66	0.66	85
LOC	0.42	0.36	0.39	58
TIM	0.53	0.37	0.43	27
Average/Total	0.56	0.51	0.53	170

Table 7.6: Classification report of BiLSTM-CRF on Caution and Advice testing dataset

### 7.1.3 Donations Dataset

60% of Donations dataset was used for training CRF and BiLSTM-CRF models, 20% for validating them, and the remaining 20% were used to appraise both models. Table 7.7 shows the values of the computed metrics for both models. The classification reports of our evaluated models CRF and BiLSTM-CRF can

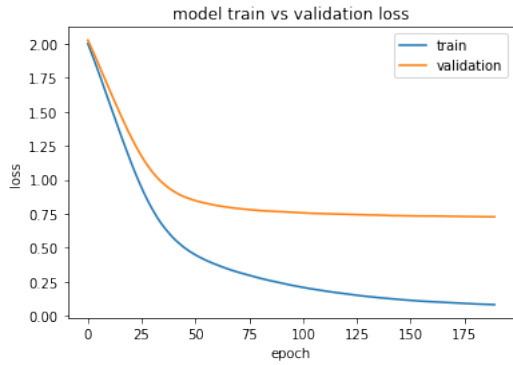


Figure 7.5: CRF loss on Donations Training and Validation Datasets

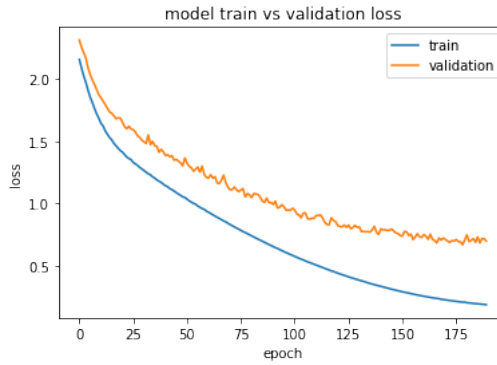


Figure 7.6: BiLSTM-CRF loss on Donations Training and Validation Datasets

be seen in Tables 7.8 and 7.9. Figures 7.5 and 7.6 show the learning curve of the loss of both models on the training and validation datasets.

<b>Metric</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
F1-score	0.336	0.255
Accuracy	0.865	0.85
Hit-ratio	0.61	0.536
Detection Rate	0.731	0.78
Bleu-Score	0.792	0.796

Table 7.7: Results of CRF and BiLSTM-CRF on Donations testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
DON	0.36	0.12	0.18	33
LOC	0.71	0.53	0.61	32
REP	0.25	0.1	0.14	31
Average/Total	0.44	0.25	0.31	96

Table 7.8: Classification report of the CRF model on Donations testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
DON	0	0	0	33
LOC	0.71	0.47	0.57	32
REP	0.27	0.13	0.17	31
Average/Total	0.32	0.2	0.24	96

Table 7.9: Classification report of BiLSTM-CRF on Donations testing dataset

#### 7.1.4 Information Source

Information Source dataset was also split into 60% for training, 20% for validating, and 20% for testing the CRF and our deep learning models. Table 7.10 shows the results of both models. Tables 7.11 and 7.12 present the classification report of the tested models. Figures 7.7 and 7.8 show the learning curve of the loss of both models on the training and validation datasets.

<b>Metric</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
F1-score	0.113	0.423
Accuracy	0.766	0.759
Hit-ratio	0.821	0.893
Detection Rate	0.928	0.982
Bleu-Score	0.767	0.78

Table 7.10: Results of CRF and BiLSTM-CRF on Information Source testing dataset

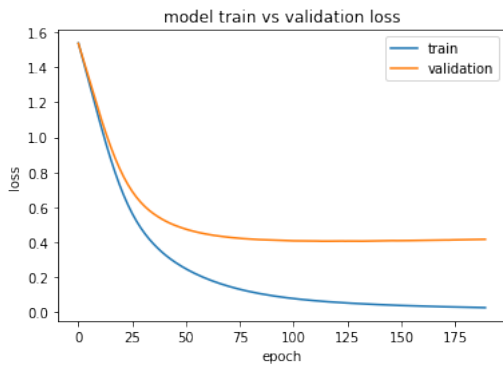


Figure 7.7: CRF loss on Information Source Training and Validation Datasets

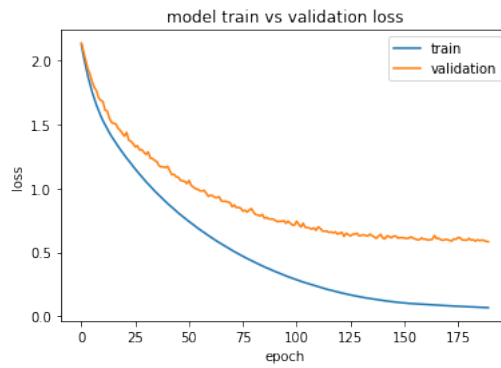


Figure 7.8: BiLSTM-CRF loss on Information Source Training and Validation Datasets

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
INC	0.12	0.17	0.14	53
SRC	0.50	0.02	0.04	50
Average/Total	0.31	0.10	0.09	103

Table 7.11: Classification report of the CRF model on Information Source testing dataset

Class	Precision	Recall	F1-score	Support
INC	0.18	0.19	0.18	53
SRC	0.65	0.70	0.67	50
Average/Total	0.41	0.44	0.42	103

Table 7.12: Classification report of BiLSTM-CRF on Information Source testing dataset

### 7.1.5 Discussion

In Table 7.1, we can see from the results that our deep learning model didn't perform better than the baseline model for Casualties and Damages dataset. The CRF model got a 0.625 for F1-score, whereas the BiLSTM-CRF scored a lower value of 0.4. Plus, the Bleu-score decreased slightly from 0.977 in the CRF model to 0.971 in the BiLSTM-CRF model. However, the values obtained for Bleu-score are considered high, and this is due to the matching process exercised by BLEU, where BLEU compares the sequence of predicted labels to the original ones for every tweet, and in the testing dataset, 98% of the words are originally tagged as 'O' and predicted correctly 825 words/840 words  $\approx$  0.98. In the testing dataset, there are only 14 tokens out of 840 initially tagged with one of the subclasses of PEO and 1 token is tagged as 'B-INF'. And, if we take a look at the classification report of the BiLSTM-CRF model in Table 7.3, the F1-score for INF class is 0 which simply means that our model didn't predict correctly the label for the word

that stands for infrastructure. This is due to having a tiny dataset which contains 138 tweets in total which are split into 110 tweets for training the model and 28 to evaluate it, wherein the 110 learning tweets we only have 23 tokens tagged with INF label. In addition to that, the accuracy also decreased from 0.988 to 0.985, and this is expected since the number of correct predictions decreased. We could also see from Table 7.1 that the hit-ratio and detection rate are alternatively low 0.178 and decreased to 0.107 and 0.143, respectively, due to having originally only 14 words in 10 different tweets to be tagged correctly as subclasses of PEO and 1 word in a separate tweet to be tagged as ‘B-INF’. Now, we have in total 11 out of 28 tweets where we could get a good hit-ratio. Thus, if we assume that we hit a correct prediction in all those 11 tweets, we would get a maximum hit-ratio equal to  $11/28 \approx 0.39$  which isn’t the case here. Moreover, if we want to calculate the ratio of words marked as one of the subclasses of PEO over the number of predicted words, we would get  $14/840 \approx 0.017$  which means that it is logical to have the hit ratio low and the detection rate low as well. Indeed, in the case of Casualties and Damages dataset, the baseline outperformed our deep learning model.

As for the Caution and Advice results, we could see that the F1-score value increased by our model from 0.49 to 0.537. Our concern as always is the Bleu-score which in this case got to 0.873 compared to 0.851 value calculated by the baseline model. We could also see a slight increase in the value of Accuracy from 0.887 to 0.895. The Hit-ratio decreased from 0.91 to 0.863 and the Detection rate



also decreased from 0.977 to 0.966. If we want to see what's behind the increase of values in F1-score, we could see in Table 7.5, that the F1-score for class 'LOC' is 0.29 which was improved by our model and got to 0.39. As for the 'TIM' class, F1-score increased from 0.32 to 0.43. There's nothing to say in this case other than our model in this dataset made a good improvement. This could be seen in Figures 7.3 and 7.4. For the baseline curves, they have the same behavior they decreased suddenly until epoch 20, after that, they started decreasing slowly until they got to fixed values 0.4 and 0.07 respectively at epoch 80. The gap between the two curves is high and both curves are parallel at that stage. However, in Figure 7.4, the training curve decreases gradually with the increase of the number of epochs, and the validation curve faces some tiny ups and downs in the error value while decreasing. The gap between the two learning curves in our model is very tight compared to the one we got by the baseline model. At epoch 200, we could see that the difference between the validation error and the training error is equal to  $0.4 - 0.05 = 0.35$  for the baseline model, whereas, it is equivalent to  $0.3 - 0.1 = 0.2$  for our BiLSTM-CRF model. This shows that our deep learning model indeed outperformed the baseline model.

Moving to the Donations dataset, we can see in Table 7.7 that our model gave a Bleu-score value of 0.796, somewhat higher than the one obtained by the baseline model which is 0.792. The detection rate also increased from 0.731 to 0.78 by the BiLSTM-CRF model. However, the F1-score decreased from 0.336 to 0.255. Plus, the Accuracy value dropped from 0.865 to 0.85 and the Hit-

ratio went down to 0.536 by our deep learning model. This implies that the tokens are not being marked properly by our system. Additionally, Table 7.9 displays 0 value for the F1-score of the DON class, which is the root cause of the reduction faced in the final F1-score computed by our model. For the other classes, our model's calculated F1-score for REP class improved slightly from 0.14 to 0.17 and decreased a little bit in value for LOC class from 0.61 to 0.57. Checking the information extracted for each tweet in Table 4.3, we could see that there's a diversity in the answers provided, where for instance, the value for "Donation" extracted is either a word like "assist" or a sequence of words like "100% of proceeds", not to mention the place of the extracted information in the original tweet, which made it hard for our BiLSTM-CRF model to retrieve the information related to donation, since it's learning different approaches compared to the small training dataset. Furthermore, while examining the 163 training tweets, we could see that there is duplication of some tweets, but with different tags linked to the words in each duplicate tweet. In this case, our system would be confused when it is being evaluated on the testing dataset. This will be clarified more in the following Error Analysis section. Lastly, we could say that the CRF model performed better than the BiLSTM-CRF model because it got a better value for F1-score ( $0.336 > 0.255$ ), although we have approximately the same Bleu-score for both models.

Finally, concerning the testing dataset of Information Source, we could see in Table 7.10 a remarkable improvement for the F1-score where it got to 0.423 by

our BiLSTM-CRF model compared to the value reached by the baseline model which is 0.113. This improvement could also be seen in the value of Hit-ratio where the Hit-ratio achieved by our model is of 0.893, whereas it reached 0.821 by the CRF model. Additionally, the Detection rate obtained by our model is relatively high 0.982 compared to the value calculated by the baseline model which is 0.928. Moreover, the Bleu-score which is the most important metric in our study increased from 0.767 to 0.78. Table 7.12 implies that we have 50 vectors or 50 sequence of words that are tagged with sub-classes of ‘SRC’ for which the F1-score got to 0.67 by our model for the SRC class, whereas, we only got a value of 0.04 by the baseline model for this class. This is the reason behind the great improvement in the final F1-score computed by our model. Furthermore, the F1-score for class INC increased from 0.14 to 0.18. If we take a look at Figures 7.7, 7.8, we could see that the loss curve for the training dataset in Figure 7.8 decreases gradually with the increase of the number of epochs and gets to a fixed value of approximately 0.1 when the number of epochs gets to 200, whereas in Figure 7.7, it decreases very fast and gets to 0.4 on epoch 30, and then starts decreasing slowly. We could see that the loss on the validation set has the same behavior as the one on the training set and the gap between these two is considered high. However, in Figure 7.8, the learning curve of the validation set shows a few ups and downs in the loss value while decreasing in general. The same behavior is seen for the learning curve of the training set, and the gap between these two is thinner than the one of the baseline. In Figure 7.8,

the validation curve is converging towards the training curve, in contrast to what is shown in Figure 7.7. Therefore, our deep learning model surpassed the baseline model for the Information Source dataset.

### 7.1.6 Error Analysis

As mentioned earlier, the baseline CRF model outperformed our BiLSTM-CRF model on the Casualties and Damages dataset. While checking the predicted labels by our deep learning model, we could see that it has mistaken in predicting some words as “O” instead of a sub-class of “PEO”. We captured the foreseen tags by both models in Tables 7.13 and 7.14 where only 2 differences appeared. These differences explain the reduction in the F1-score of the BiLSTM-CRF model, where the value of True Positive decreased by 1 due to labeling “122” as “O” (see Table 7.14), and the value of False Positive increased by 1 since “church” got tagged as “B-PEO” by our model (check Table 7.13). We also checked in parallel the training dataset, and we identified the reason behind predicting “122” as “O” instead of “B-PEO”. The BiLSTM layer obliges our model to learn from sequence of words provided in the learning data, where it was taught for instance to label the word that comes strictly before “in Joplin, Missouri,” as “O” and the second word that comes after “that” to be tagged as “O”. Those mistakes are related to the limited learning dataset (110 tweets, where only 23 out of 3300 words are tagged with sub-classes of INF, and 52 out of 3300 words are labeled as one of the

sub-classes of PEO). We can conclude that BiLSTM-CRF would have performed better if we had a training dataset rich in words tagged with sub-classes of PEO and INF.

<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
@andersoncooper:	O	O	O
I	O	O	O
took	O	O	O
this	O	O	O
earlier.	O	O	O
Cross	O	O	O
still	O	O	O
stands	O	O	O
above	O	O	O
destroyed	O	O	O
church	B-INF	B-INF	B-PEO
#Joplin	O	O	O
<a href="http://yfrog.com/h2d7rgrj">http://yfrog.com/h2d7rgrj</a>	O	O	O

Table 7.13: Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Casualties and Damages testing dataset

<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
@iamjonathancook:	O	O	O
Prayersforjoplin	O	O	O
@cnnbrk:	O	O	O
#Tornado	O	O	O
that	O	O	O
killed	O	O	O
122	B-PEO	B-PEO	O
in	O	O	O
Joplin,	O	O	O
Missouri,	O	O	O
was	O	O	O
EF-5	O	O	O
with	O	O	O
top	O	O	O
winds	O	O	O
of	O	O	O
200+	O	O	O
mph	O	O	O
htt	O	O	O

Table 7.14: Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Casualties and Damages testing dataset

In the Caution and Advice testing dataset, we could see some words that should be tagged as ‘I-LOC’, got predicted as either ‘O’, ‘B-LOC’, or ‘E-LOC’ by our model. Plus, some words that refer to the time of the incident that should be labeled as ‘B-TIM’, ‘I-TIM’, or ‘E-TIM’ were predicted as ‘O’. In addition, we can see only a few mistakes when predicting the words that refer to an incident as ‘O’. Moreover, most of the words that should be predicted as ‘O’ got associated with one of the three-class labels. Table 7.15 shows the differences between the predictions of the baseline and our model on the testing dataset for one of the evaluated tweets. The table explains that ‘from’ should be tagged as ‘B-TIM’; however, our model predicted it as ‘O’. We checked in parallel the training dataset, and we could see that for the sequence of word that occurs in this example, the model learned from the training dataset that ‘from’ should be tagged ‘O’, according to the following sequence of words ‘Tornado Watch from 5/24/2011 9:12 PM to 10:00 PM EDT’ that is present in the training dataset, where each token was joined to the corresponding tag in the subsequent series of terms ‘B-INC E-INC B-TIM I-TIM I-TIM I-TIM I-TIM I-TIM I-TIM E-TIM’. Therefore, we could see when our model was tested on the testing dataset, the Bi-LSTM layer helped in recognizing the same structure of words and predicted the tags according to what it has learned previously. However, when it got to predicting the location, for the sequence of words ‘for Ray County,’ we thought that the model learned from the training dataset to tag the first word after ‘for’ as ‘B-LOC’ in case it was followed by a comma. In this example, ‘Ray’ occurred

in the training dataset with an ‘O’ tag. Thus, we concluded that ‘County,’ got predicted as ‘B-LOC’ instead of ‘I-LOC’ from what it has previously learned. In this case, we could only say that if we had more training dataset, our model would have even performed better and got a Bleu-score higher than 0.873.



<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
New	O	O	O
event.	O	O	O
Tornado	B-INC	B-INC	B-INC
Warning	E-INC	E-INC	E-INC
from	B-TIM	O	O
5/25/2011	I-TIM	O	B-TIM
1:28	I-TIM	B-TIM	I-TIM
PM	I-TIM	I-TIM	I-TIM
to	I-TIM	I-TIM	I-TIM
2:00	I-TIM	I-TIM	I-TIM
PM	I-TIM	I-TIM	I-TIM
CDT	E-TIM	E-TIM	E-TIM
for	O	O	O
Ray	B-LOC	O	O
County,	I-LOC	I-LOC	B-LOC
Carroll	I-LOC	I-LOC	I-LOC
County	E-LOC	E-LOC	E-LOC
.	O	O	O
More	O	O	O
information....	O	O	O
<a href="http://fb.me/Ix0gvwIa">http://fb.me/Ix0gvwIa</a>	O	O	O

Table 7.15: Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Caution and Advice testing dataset

Now let's move to Donations' error analysis. We could see that for some words that should be marked as one of the subclasses of 'REP', they got labeled as 'O', other words that are truly labeled with 'O', got a prediction of one of the subclasses of 'DON' or 'REP', and words that are truly linked to one of the subclasses of 'DON' got predicted as 'O'. One of these tweets is shown in table 7.16 which shows the difference between the labels predicted by our model and the baseline model for the same tweet. Both models are predicting either different or the same label as the Ground Truth, where Ground Truth stands for the expected prediction. We checked in parallel the training dataset just to compare why some words are being predicted as 'O', and we could see that for instance in Table 7.16, 'Circus elephants' should be tagged as 'B-REP E-REP', however, it was labeled by our both model as 'O O'. It looks like our BiLSTM-CRF model learned from the training dataset to tag the words that appear before the series of tokens 'help with Tornado' as 'O'. 'help' was linked to 'B-DON' because the occurrence of this word followed by 'with Tornado' was learned in the training dataset to be labeled as 'B-DON', but the baseline model labeled it correctly as 'O' because CRF doesn't check previous and next tokens as the BiLSTM does. For the rest of the tweet, we could see that the sequence 'Tornado Clean Up' should be predicted as 'B-DON I-DON E-DON', though they were all tagged as 'O' by our model since it marked 'help' with 'B-DON', then the following words should be tagged as 'O' as it has learned from the training dataset. In addition to all of this, we noticed some duplicate tweets in the training dataset where the

tags associated with words of the duplicate tweets are different. For instance, for the same series of tokens ‘Txt REDCROSS to’ for the same tweet in the training dataset, we have 3 different tag sequences ‘O B-REP O’, ‘O O O’, and ‘B-DON I-DON I-DON’. When running our system on the testing dataset, we faced the same sequence of words that should have been predicted as ‘B-DON I-DON I-DON’, but it was unfortunately predicted by our model as ‘O B-REP O’. This variety of answers and limited training dataset has led to confusion in our model and made it hard for our system to get better results compared to the baseline model.

Word	Ground Truth	CRF	BiLSTM-CRF
Ugh,	O	O	O
deplorable.	O	O	O
@majornelson:	O	O	O
Circus	B-REP	O	O
elephants	E-REP	O	O
help	O	O	B-DON
with	O	O	O
Tornado	B-DON	O	O
Clean	I-DON	O	O
Up	E-DON	O	O
in	O	O	O
Joplin,	B-LOC	B-LOC	B-LOC
MO	E-LOC	E-LOC	E-LOC
<a href="http://t.co/01qIMuM">http://t.co/01qIMuM</a>	O	O	O

Table 7.16: Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Donations testing dataset

Lastly, for the testing dataset of Information Source, we could see that most of the words that should be tagged as ‘B-INC’, got predicted by our system as either ‘O’ or ‘I-INC’. In addition, some words labeled as ‘O’ got predicted by our model as either ‘B-SRC’, ‘B-INC’, or ‘I-INC’. However, for most of the words that are

labeled as ‘B-SRC’, they were predicted correctly by our model, just in few cases we got a prediction of ‘O’ for some tokens labeled as ‘B-SRC’. We show a sample of a tweet that presents such cases in table 7.17. We can see that ‘cutting across the Connecticut’ was predicted by both models as ‘I-INC I-INC I-INC I-INC E-INC’, whereas they should have been predicted as ‘E-INC O O O O’. For our BiLSTM-CRF model, it checks the previous and next words to tag the current tokens correctly based on what it has learned from the training dataset. Checking the training dataset, our analysis for this example would be that it has learned that ‘across’ in such cases should be preceded and succeeded by words that are tagged as ‘I-INC’, so ‘cutting’ got tagged as ‘I-INC’ and ‘the’ as well, and since it has acquired the following tags ‘I-INC I-INC I-INC I-INC I-INC’ for this sequence of words ‘video of a tornado hitting’, our series of words ‘video of a tornado’ was marked as ‘I-INC I-INC I-INC I-INC’, and ‘River’ was learned to be tagged as ‘O’ in this sequence of words, thus ‘Connecticut’ got labeled as ‘E-INC’. In addition, while checking the information extracted of the tweets, we could see that 90% of the tweets’ incident extracted is a copy-paste of the whole tweet excluding the URLs that usually refer to the information source. This made it hard for our both models to tag the incident words correctly since we only have 224 tweets for training and the patterns to be learned from these tweets are various compared to the limited number of learning tweets. In Table 7.18, we can see that the incident extracted from the second tweet is a copy of it. In addition, the incidents extracted for the first and last tweet are a copy of the same tweet excluding the

last token which is ‘#despicable.’ and ‘http://vimeo.com/24290944’, respectively. Therefore, the first word of the extracted information for the incident would be tagged as ‘B-INC’, the last word as ‘E-INC’, and all the words that fall in-between would be tagged as ‘I-INC’. This diversity in patterns hardened the evaluation for the CRF and BiLSTM-CRF to predict correctly the tokens that are truly tagged as sub-classes of ‘INC’.

<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>BiLSTM-CRF</b>
Close-up	B-INC	B-INC	B-INC
video	I-INC	I-INC	I-INC
of	I-INC	I-INC	I-INC
a	I-INC	I-INC	I-INC
tornado	E-INC	I-INC	I-INC
cutting	O	I-INC	I-INC
across	O	I-INC	I-INC
the	O	I-INC	I-INC
Connecticut	O	E-INC	E-INC
River	O	O	O
http://itv.co/kb1LLS	B-SRC	O	B-SRC

Table 7.17: Ground Truth vs Predicted Value by CRF and BiLSTM-CRF for the Information Source testing dataset

Tweet	Incident
My cop friend volunteered in #Joplin last wknd; she said people are driving there from Ark., Ill. & Cal. to loot! #despicable.	My cop friend volunteered in #Joplin last wknd; she said people are driving there from Ark., Ill. & Cal. to loot!
Back at my desk after the tornado warning. That was fun. #sarcasm #thingsyoudontsayinNYC	Back at my desk after the tornado warning. That was fun. #sarcasm #thingsyoudontsayinNYC
I just uploaded "VIOLENT Dibble/Washington/Goldsby Tornado! May 24, 2011" on Vimeo: <a href="http://vimeo.com/24290944">http://vimeo.com/24290944</a>	I just uploaded "VIOLENT Dibble/Washington/Goldsby Tornado! May 24, 2011" on Vimeo:

Table 7.18: Sample of Information Source tweets Extracted Information

## 7.2 Evaluation of Deep Learning on FA-KES Dataset

The same procedure was applied to the FA-KES dataset. However, the BiLSTM-CRF model with an attention layer on top of it (i.e. OpenTag) performed better than BiLSTM-CRF. To check how well our deep learning model is doing, we created the same baseline (CRF) model which was used in our previous experiments on the Joplin dataset. Both models CRF and OpenTag were trained and tested on the same data.

As for the experiments, we created three subsets of data from the FA-KES articles: articles' titles, articles' titles and contents, and articles' titles and first paragraphs. CRF and OpenTag were trained on the training data and tested on the corresponding data of each subset. Both models were trained for 200 epochs with a batch of size 32 and used RMSProp as an optimizer. We measured F1-score and compared the results obtained on the subsets. Additionally, we drew

the learning curve of the loss of the training and validation datasets for each of the evaluated models.

### 7.2.1 FA-KES Titles Dataset

We split the titles in the FA-KES dataset into 60% for training, 20% for validating, and 20% for testing the models. We trained the CRF and OpenTag models on the learning data and experimented them on the testing dataset. Tables 7.23 and 7.24 present the classification report of our evaluated models. Figures 7.9 and 7.10 show the learning curve of the loss of both models on the training and validation datasets. Figures 7.11, 7.12, 7.13, and 7.14 present the attention executed by OpenTag on some examples of the FA-KES titles testing data.

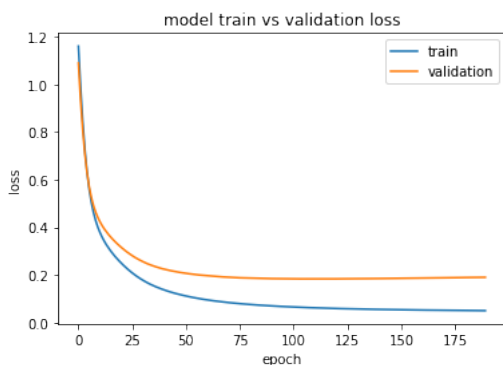


Figure 7.9: CRF loss on FA-KES Titles Training and Validation Datasets

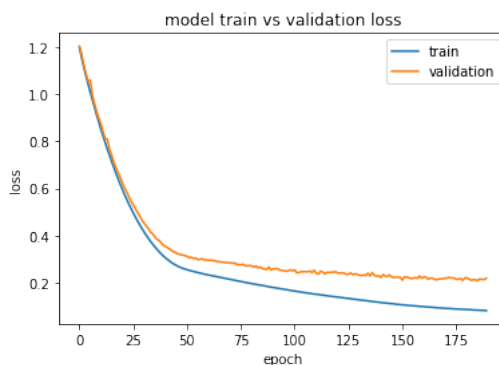


Figure 7.10: OpenTag loss on FA-KES Titles Training and Validation Datasets



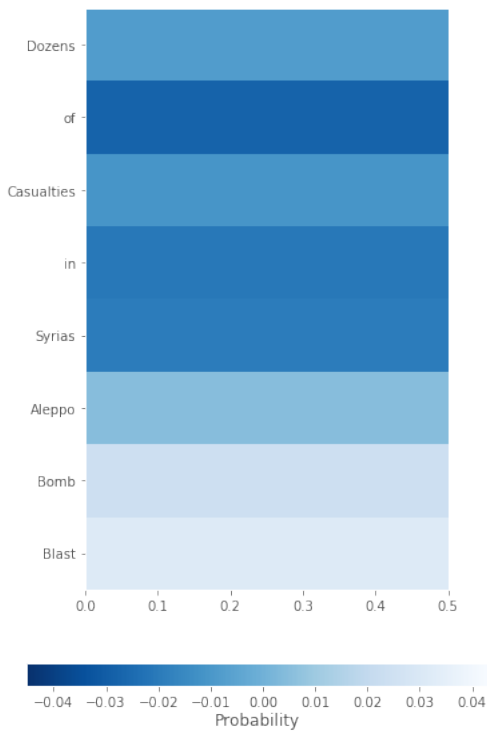


Figure 7.11: Example of Attention Visualization on FA-KES Titles testing dataset

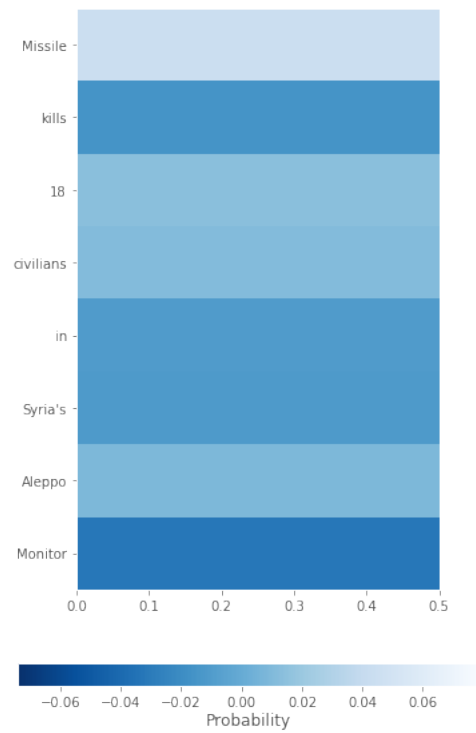


Figure 7.12: Example of Attention Visualization on FA-KES Titles testing dataset

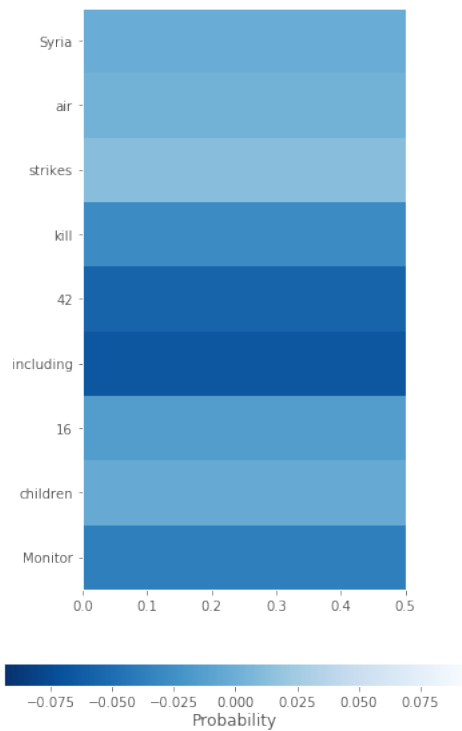


Figure 7.13: Example of Attention Visualization on FA-KES Titles testing dataset

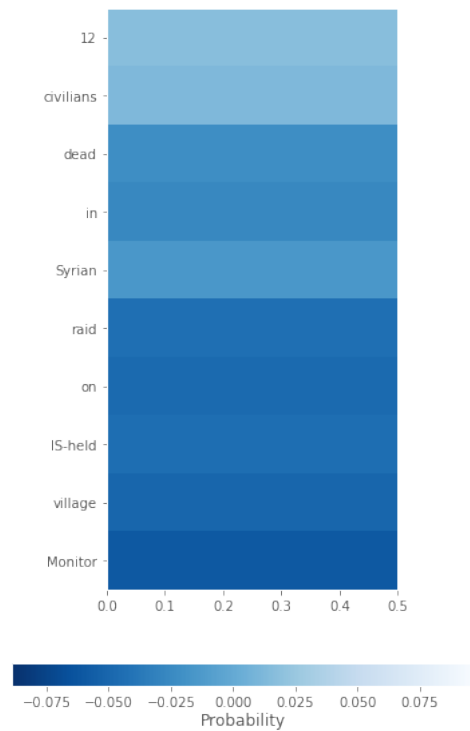


Figure 7.14: Example of Attention Visualization on FA-KES Titles testing dataset

<b>Word</b>	<b>Ground Truth</b>	<b>OpenTag</b>
Dozens	O	O
of	O	O
casualties	O	O
in	O	O
Syrias	O	O
Aleppo	B-LOC	B-LOC
Bomb	B-COD	B-COD
Blast	E-COD	E-COD

Table 7.19: Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.11

<b>Word</b>	<b>Ground Truth</b>	<b>OpenTag</b>
Missile	B-COD	B-COD
kills	O	O
18	B-CIV	B-CIV
civilians	E-CIV	E-CIV
in	O	O
Syria's	O	O
Aleppo	B-LOC	B-LOC
Monitor	O	O

Table 7.20: Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.12

<b>Word</b>	<b>Ground Truth</b>	<b>OpenTag</b>
Syria	O	O
air	B-COD	B-COD
strikes	E-COD	E-COD
kill	O	O
42	O	O
including	O	O
16	B-CHD	B-CHD
children	B-CHD	E-CHD
Monitor	O	O

Table 7.21: Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.13

<b>Word</b>	<b>Ground Truth</b>	<b>OpenTag</b>
12	B-CIV	B-CIV
civilians	E-CIV	E-CIV
dead	O	O
in	O	O
Syrian	O	O
raid	O	O
on	O	O
IS-held	O	O
village	O	O
Monitor	O	O

Table 7.22: Ground Truth vs Predicted Value by OpenTag for the example shown in Figure 7.14

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
ACT	0.53	0.57	0.55	61
COD	0.47	0.56	0.51	81
LOC	0.77	0.77	0.77	111
CIV	0.44	0.49	0.46	47
NCV	0.63	0.49	0.55	35
CHD	1.00	0.50	0.67	6
Average/Total	0.60	0.61	0.60	341

Table 7.23: Classification report of the CRF model on FA-KES Titles testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
ACT	0.55	0.52	0.54	61
COD	0.49	0.56	0.52	81
LOC	0.73	0.70	0.72	111
CIV	0.44	0.60	0.50	47
NCV	0.38	0.43	0.41	35
CHD	0.80	0.67	0.73	6
Average/Total	0.57	0.59	0.58	341

Table 7.24: Classification report of OpenTag on FA-KES Titles testing dataset

## 7.2.2 FA-KES Titles and First Paragraphs Dataset

After collecting the first paragraphs from the FA-KES articles dataset, we concatenated the title and first paragraph for every article in the news dataset. We split this newly created subset into 60% for training, 20% for validation, and 20% for testing the models. We trained the CRF and OpenTag models on the training dataset and evaluated them on the testing dataset. Tables 7.25 and 7.26 show the classification report of CRF and OpenTag, respectively. Figures 7.15 and 7.16 show the learning curve of the loss of both models on the training and validation datasets.

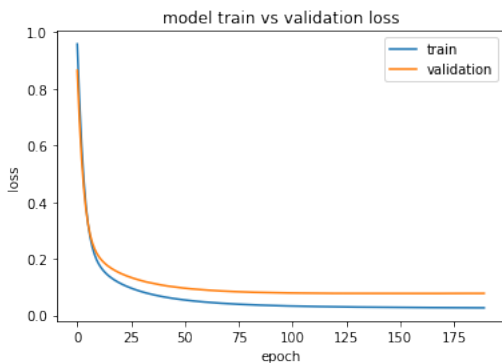


Figure 7.15: CRF loss on FA-KES Titles and First Paragraphs Training and Validation Datasets

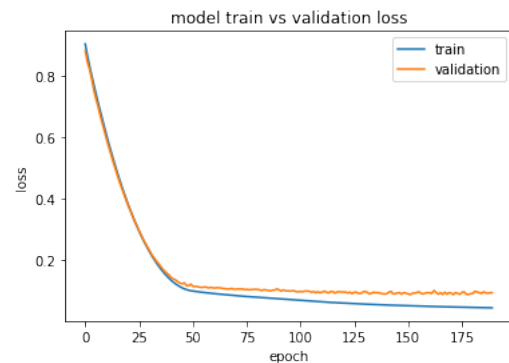


Figure 7.16: OpenTag loss on FA-KES Titles and First Paragraphs Training and Validation Datasets

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
ACT	0.52	0.34	0.41	147
COD	0.44	0.43	0.44	148
LOC	0.73	0.78	0.75	236
CIV	0.58	0.47	0.52	127
NCV	0.39	0.38	0.39	39
CHD	0.59	0.70	0.64	23
WMN	0.00	0.00	0.00	6
Average/Total	0.57	0.54	0.55	726

Table 7.25: Classification report of the CRF model on FA-KES Titles and First Paragraphs testing dataset



<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
ACT	0.41	0.38	0.40	147
COD	0.39	0.39	0.39	148
LOC	0.62	0.76	0.68	236
CIV	0.62	0.48	0.54	127
NCV	0.24	0.23	0.23	39
CHD	0.50	0.26	0.34	23
WMN	0.00	0.00	0.00	6
Average/Total	0.50	0.51	0.50	726

Table 7.26: Classification report of OpenTag on FA-KES Titles and First Paragraphs testing dataset

### 7.2.3 FA-KES Titles and Contents Dataset

We noticed that the content of every article in the FA-KES dataset doesn't include the title of the article. Thus, we created a new subset that is composed of the title and the content of each article. We split this newly created subset into 60% for training, 20% for validation, and 20% for testing the models. We trained the CRF and OpenTag models on the training dataset and examined them on the testing dataset. Tables 7.27 and 7.28 display the classification report of CRF and OpenTag, respectively. Figures 7.17 and 7.18 show the learning curve of the loss of both models on the training and validation datasets.

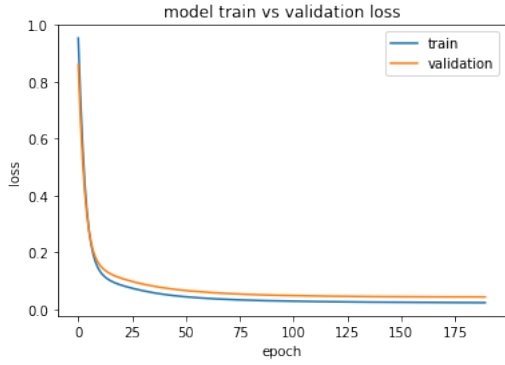


Figure 7.17: CRF loss on FA-KES Titles and Contents Training and Validation Datasets

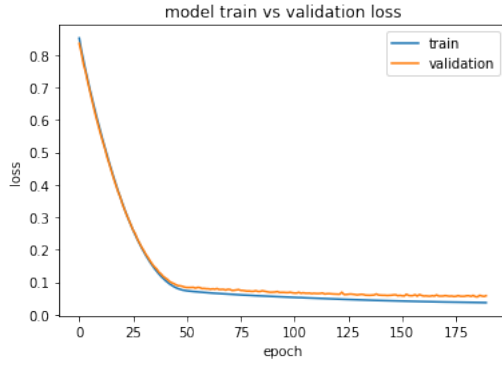


Figure 7.18: OpenTag loss on FA-KES Titles and Contents Training and Validation Datasets

Class	Precision	Recall	F1-score	Support
ACT	0.37	0.08	0.14	289
COD	0.37	0.11	0.17	309
LOC	0.66	0.74	0.70	686
CIV	0.52	0.24	0.33	190
NCV	0.54	0.36	0.43	61
CHD	0.55	0.25	0.34	44
WMN	0.00	0.00	0.00	10
DAT	0.68	0.48	0.57	124
Average/Total	0.53	0.41	0.44	1713

Table 7.27: Classification report of the CRF model on FA-KES Titles and Contents testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
ACT	0.30	0.12	0.17	289
COD	0.22	0.01	0.02	309
LOC	0.63	0.66	0.65	686
CIV	0.40	0.12	0.18	190
NCV	0.00	0.00	0.00	61
CHD	0.00	0.00	0.00	44
WMN	0.00	0.00	0.00	10
DAT	0.48	0.37	0.42	124
Average/Total	0.42	0.33	0.34	1713

Table 7.28: Classification report of OpenTag on FA-KES Titles and Contents testing dataset

## 7.2.4 Discussion

As a summary of our findings, CRF performed better than OpenTag and especially on the FA-KES Titles subset where it scored the highest value for F1-score which is 0.60. In Table 7.23, we can see that the average F1-score is 0.6 whereas the OpenTag model scored 0.58 as F1-score. This highlights the fact that the baseline model CRF performed slightly better than our deep learning model. If we want to take a closer look at the results, we can see that for LOC (i.e. location of the incident), CRF returned 0.77 as F1-score and OpenTag gave a 0.72 score.

Additionally, NCV (i.e. non-civilians who died in the incident) scored 0.55 as F1-score by the baseline paradigm, whereas it scored only 0.41 by our model. However, OpenTag returned better results than the CRF for CHD, CIV, and COD. CRF and OpenTag models returned approximately the same results for COD and ACT classes, but in general, CRF outperformed OpenTag since F1-score is 0.60 which is greater than the one returned by OpenTag 0.58. This could be seen also in Figures 7.9 and 7.10. The baseline curves decreased suddenly until epoch 10, then started to decrease slowly until they got to stable values which are 0.2 and 0.05 for the validation and training curves respectively. We can see that the gap between the baseline curves is stable starting epoch 100 which is equal to  $0.2 - 0.05 = 0.15$ . Now, if we take a look at Figure 7.10, we can see that there were some tiny ups in the loss score for the validation set while both curves were generally decreasing gradually until epoch 40. The training curve kept decreasing until epoch 200; however, there were some tiny ups and downs in the validation curve until it reached epoch 200 where the gap between the learning curves is equal also to  $0.2 - 0.05 = 0.15$ . Although the same gap can be seen in both Figures 7.9 and 7.10 but if we compile the model for more than 200 epochs, the gap will increase between the curves for OpenTag model because the loss on the Validation dataset is increasing while the loss on the training dataset is decreasing which is not the case in the baseline model. CRF performed better than OpenTag on FA-KES Titles dataset. Furthermore, Figures 7.11, 7.12, 7.13, and 7.14 show the attention performed by our model on all the words that

appeared in some of the titles in the testing dataset. Under each one of the visualization graphs, the higher the probability value, the lighter is the color, in other words, the color is brighter for the tokens that our model concentrated on. The color bar seen under each one of the mentioned figures represents the color map scaling. Table 7.19 presents the tags predicted by OpenTag for each word that appears in the title. We could see for this first example the predicted values match the Ground Truth column where ‘Aleppo’ was tagged as ‘B-LOC’, and ‘Bomb Blast’ as ‘B-COD E-COD’. This means that our model focused more on these 3 words among all the other words that appeared in the title and this could be seen from Figure 7.11 where the attention on ‘Aleppo’ is highlighted by Blue light color, ‘Bomb’ and ‘Blast’ by color leaning to white. If we move to the next example, Table 7.20 presents the labels predicted by our deep learning model that also match the Ground Truth column values and the attention on these words ‘Missile’, ‘18 civilians’ and ‘Aleppo’ is presented by a lighter color than the other words in Figure 7.12. Moreover, the attention exercised by OpenTag is highlighted by lighter colors for the sequence of words ‘air strikes’ (predicted as the cause of death) and ‘16 children’ (predicted as number of dead children) in Figure 7.13 for the example of title shown in Table 7.21. Lastly, ‘12 civilians’ which was predicted as number of dead civilians by our model in Table 7.22 got the highest attention among all the words referring to Figure 7.14.

Moving to the FA-KES Titles and First Paragraphs subset, we can see in Table 7.25 that the baseline model returned 0.55 as F1-score which is higher than

the score of OpenTag which is 0.50 as it is shown in Table 7.26. A remarkable improvement in the F1-score for CHD (i.e. the number of dead children) can be viewed in Table 7.35 where it scored 0.64 compared to the result obtained by OpenTag which is 0.34. For all the classes except the CIV (i.e. the number of dead civilians), CRF performed better than the OpenTag model, returning a higher score for F1-score on average. Plus, if we contemplate Figures 7.15 and 7.16, we can see the same behavior of both models like the one we saw earlier for the FA-KES Titles dataset. The baseline curves decreased suddenly until they reached epoch 10, then they started decreasing slowly until the two curves became parallel maintaining the same value starting epoch 100 where the gap between them remained the same ( $0.15 - 0.10 = 0.05$ ). As for OpenTag loss on the validation and training datasets, we can see that both curves decreased gradually until they reached epoch 40, then the training curve continued decreasing slowly whereas the validation continued increasing, to end up having the same gap between both curves ( $0.15 - 0.10 = 0.05$ ). Eventually, the gap will increase if we continue training the model for more than 200 epochs which leaves the CRF as the winner over the OpenTag model.

Finally, regarding the performance of CRF and OpenTag on FA-KES Titles and Contents testing dataset, we could see from Tables 7.27 and 7.28 that CRF scored better than OpenTag for F1-score ( $0.44 > 0.34$ ). This is explained by having higher F1-score value for several classes including COD, LOC, CIV. Moreover, we can see that for NCV and CHD classes, the baseline returned 0.43 and 0.34

respectively as F1-score, whereas OpenTag scored 0 for both classes and this is due to having 676 words out of 517600 labeled with ‘NCV’ tag and 324 tokens out of 517600 marked with ‘CHD’ tag. Only for ACT class, OpenTag returned better F1-score ( $0.17 > 0.14$ ). Lastly, both Figures 7.17 and 7.18 show the same behavior seen for FA-KES Titles and First Paragraphs subset. The gap between the curves is very small; however, it tends to increase after epoch 200 for the OpenTag model, whereas it remains the same for the CRF paradigm. Therefore, CRF performed better than OpenTag on FA-KES Titles and Contents subset.

### 7.2.5 Error Analysis

As discussed, CRF performed better than the OpenTag model on the FA-KES titles subset. If we take a look back at Tables 7.23 and 7.24 we could see that for CIV class our model got a higher F1-score than the baseline ( $0.50 > 0.46$ ) and this is illustrated in Table 7.30 where CRF didn’t tag ‘15 Medics’ with the correct labels whereas OpenTag has learned from the training dataset that the words that come strictly before ‘Killed’ should be tagged as ‘B-CIV E-CIV’. This would be credited to the fact that OpenTag has a BiLSTM layer in its architecture. Additionally, we noticed that for NCV class, OpenTag didn’t perform as well as it did for other classes. It only scored 0.41 while CRF got a 0.55 F1-score for NCV class. Table 7.29 shows an example for one of the titles in the FA-KES Titles testing dataset where ‘Nusra’ was labeled as ‘E-NCV’ instead of ‘I-NCV’

by OpenTag and this could be explained by checking the training dataset where the BiLSTM layer was taught to tag the words that come after ‘kills’ token as ‘B-NCV I-NCV E-NCV’ until it got to ‘Front’ which should be tagged as ‘O’ according to the training dataset especially that the ‘-’ sign should be preceded and followed by words labeled as ‘O’. One thing to note here, the training dataset only contains 257 words out of 19410 marked with NCV class which explains why our neural network model wasn’t able to tag all the words related to that class correctly scoring only 0.41 as F1-score for NCV.

As for the FA-KES Titles and First Paragraphs subset, we can see in Tables 7.25 and 7.26 that CRF gave better scores than the OpenTag for all classes. For the number of women who were killed during the incidents, we can see that both paradigms weren’t able to detect any word related to WMN class and this is because of having only 27 words out of 129400 tagged with WMN in the training dataset. A mini-example of the FA-KES Titles and First Paragraphs testing dataset is shown in Table 7.31 where ‘4 children’ were tagged correctly by CRF as ‘B-CHD E-CHD’ and weren’t detected by our deep learning model. Moreover, there are only 155 words out of 129400 tagged with CHD in the learning dataset which is not enough for our deep learning model to learn from. In this example, the sequence of words ‘terrorist attack’ that represent the cause of death was labeled correctly by our model because it has learned from the training dataset that those two words should be marked as ‘B-COD’ and ‘E-COD’ respectively. Having more than 100 words per article reduces the probability for our OpenTag



model to tag all the words correctly especially that we have diversity in patterns compared to the small training dataset.

Finally, Tables 7.27 and 7.28 show the classification report of CRF and OpenTag respectively on FA-KES Titles and Contents testing dataset for which CRF performed better than OpenTag. We could see in Table 7.32 that ‘airstrike’ got labeled as ‘E-ACT’ instead of ‘B-COD’ and this is due to the learning phase from which it has learned that the word that comes strictly after ‘Russian’ should be labeled as ‘E-ACT’ if the word that comes after ‘airstrike’ is ‘on’ which is the case here. Furthermore, we could see that ‘8 civilians’ got predicted as ‘O O’ instead of ‘B-CIV E-CIV’ and this is because the model was trained to tag the sequence of words that comes after ‘-’ sign as ‘O’. However, we could see for instance that the location ‘Syria’ got predicted as ‘O’ by both models instead of ‘B-LOC’. This is due to the learning dataset where ‘Syria’ appeared 1117 times in the learning dataset and most of them were tagged as ‘O’. Having more than 500 words per testing article hardened the extraction of words, in other words, it made it hard for our model to tag all the words correctly especially that we have several patterns to be learned on a limited subset which is composed of 482 articles.

<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>OpenTag</b>
Suicide	B-COD	O	B-COD
bomb	I-COD	B-COD	I-COD
attack	E-COD	E-COD	E-COD
in	O	O	O
Syria's	O	O	O
Idlib	B-LOC	B-LOC	B-LOC
mosque	O	E-LOC	B-COD
kills	O	O	O
25	B-NCV	B-NCV	B-NCV
senior	I-NCV	I-NCV	I-NCV
Nusra	I-NCV	I-NCV	E-NCV
Front	I-NCV	I-NCV	O
leader	E-NCV	E-NCV	O
-	O	O	O
Daily	O	O	O
Sabah	O	O	O

Table 7.29: Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles testing dataset

<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>OpenTag</b>
15	B-CIV	O	B-CIV
Medics	E-CIV	O	E-CIV
Killed	O	O	O
in	O	O	O
Russian	B-ACT	B-ACT	B-ACT
Airstrikes	B-COD	B-COD	B-COD
on	O	O	O
Hospital	O	O	O
in	O	O	O
Southern	O	O	O
Aleppo	B-LOC	B-LOC	B-LOC
Syrian	O	O	B-COD
Coalition	O	O	E-COD
Describes	O	O	O
the	O	O	O
Attack	O	O	O
War	O	O	O
Crime	O	O	O

Table 7.30: Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles testing dataset

<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>OpenTag</b>
4	B-CHD	B-CHD	O
children	E-CHD	E-CHD	O
and	O	O	O
a	B-WMN	O	O
woman	E-WMN	O	O
,	O	O	O
while	O	O	O
eight	O	O	O
persons	O	O	O
were	O	O	O
injured	O	O	O
in	O	O	O
terrorist	B-COD	O	B-COD
attack	E-COD	O	E-COD

Table 7.31: Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles and First Paragraphs testing dataset

<b>Word</b>	<b>Ground Truth</b>	<b>CRF</b>	<b>OpenTag</b>
Published	O	O	O
June	O	O	O
4	O	O	O
2016	O	O	O
-	O	O	O
8	B-CIV	B-CIV	O
civilians	E-CIV	E-CIV	O
killed	O	O	O
by	O	O	O
Russian	B-ACT	B-ACT	O
airstrike	B-COD	B-COD	O
in	O	O	O
Syria	B-LOC	O	O
-	O	O	O
A	O	O	O
Russian	B-ACT	B-ACT	B-ACT
airstrike	B-COD	B-COD	E-ACT
on	O	O	O
Syrian	B-LOC	O	B-ACT
city	O	O	O
Aleppo	O	B-LOC	B-LOC

Table 7.32: Ground Truth vs Predicted Value by CRF and OpenTag for the FA-KES Titles and Contents testing dataset

## 7.3 Evaluation of FA-KES Classifiers

To see how well the classifiers are doing, we created baseline classifiers that were trained on the text solely of the training subsets and tested them on the same testing dataset (excluding labels) on which the below classifiers were examined.

We will mainly focus on the value obtained for F1-score.

### 7.3.1 Actor Classifier

As mentioned before, we created three classifiers for attribute actor: Logistic Regression, LinearSVC, and Random Forest Classifier. We will see in the following subsections the results of the best performing one for each dimension.

#### 7.3.1.1 FA-KES Titles Dataset

Logistic Regression gave the best results among the three actor classifiers. Tables 7.33 and 7.34 show the classification report of our classifier on the articles' titles (excluding tags) and on the articles' titles with the labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Al-Nusra Front	0.00	0.00	0.00	3
Armed opposition groups	0.60	0.43	0.50	7
International coalition forces	0.58	0.78	0.67	9
Other	0.64	0.55	0.59	66
Russian troops	0.36	0.45	0.40	11
Self administration forces	0.67	0.67	0.67	3
Syrian government and affiliated militias	0.66	0.77	0.71	53
The organization of the Islamic State in Iraq and the Levant - ISIS	0.57	0.40	0.47	10
Average/Total	0.61	0.60	0.60	162

Table 7.33: Classification report of the baseline Logistic Regression Actor classifier on FA-KES Titles testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Al-Nusra Front	0.00	0.00	0.00	3
Armed opposition groups	0.67	0.57	0.62	7
International coalition forces	0.50	0.78	0.61	9
Other	0.64	0.55	0.59	66
Russian troops	0.43	0.55	0.48	11
Self administration forces	1.00	0.67	0.80	3
Syrian government and affiliated militias	0.63	0.72	0.67	53
The organization of the Islamic State in Iraq and the Levant - ISIS	0.67	0.60	0.63	10
Average/Total	0.61	0.61	0.61	162

Table 7.34: Classification report of Logistic Regression Actor classifier on FA-KES Titles testing dataset with tags predicted by OpenTag

### 7.3.1.2 FA-KES Titles and First Paragraphs Dataset

In this case, Logistic Regression also performed better than the other actor classifiers. Tables 7.35 and 7.36 show the classification report of our classifier on the articles' titles and first paragraphs (excluding tags) and on the articles' titles concatenated with first paragraphs and their labels predicted by our deep learning model respectively.



<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Al-Nusra Front	0.00	0.00	0.00	5
Armed opposition groups	0.75	0.33	0.46	9
International coalition forces	0.94	0.94	0.94	18
Other	0.73	0.82	0.77	39
Russian troops	0.75	0.79	0.77	19
Self administration forces	1.00	0.71	0.83	7
Syrian government and affiliated militias	0.77	0.83	0.80	52
The organization of the Islamic State in Iraq and the Levant - ISIS	0.73	0.85	0.79	13
Average/Total	0.76	0.78	0.76	162

Table 7.35: Classification report of the baseline Logistic Regression Actor classifier on FA-KES Titles and First Paragraphs testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Al-Nusra Front	0.00	0.00	0.00	5
Armed opposition groups	0.67	0.44	0.53	9
International coalition forces	0.94	0.94	0.94	18
Other	0.76	0.79	0.77	39
Russian troops	0.74	0.74	0.74	19
Self administration forces	1.00	0.71	0.83	7
Syrian government and affiliated militias	0.78	0.87	0.82	52
The organization of the Islamic State in Iraq and the Levant - ISIS	0.73	0.85	0.79	13
Average/Total	0.76	0.78	0.77	162

Table 7.36: Classification report of Logistic Regression Actor classifier on FA-KES Titles and First Paragraphs testing dataset with tags predicted by OpenTag

### 7.3.1.3 FA-KES Titles and Contents Dataset

Logistic Regression performed better than the other actor classifiers. Tables 7.37 and 7.38 show the classification report of our classifier on the articles' titles and contents (excluding tags) and on the articles' titles concatenated with their content and the labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Al-Nusra Front	0.00	0.00	0.00	1
Armed opposition groups	0.67	0.17	0.27	12
International coalition forces	0.70	0.64	0.67	11
Other	0.77	0.66	0.71	61
Russian troops	0.50	0.53	0.52	15
Self administration forces	0.33	0.33	0.33	3
Syrian government and affiliated militias	0.57	0.65	0.61	46
The organization of the Islamic State in Iraq and the Levant - ISIS	0.32	0.54	0.40	13
Average/Total	0.63	0.59	0.59	162

Table 7.37: Classification report of the baseline Logistic Regression Actor classifier on FA-KES Titles and Contents testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Al-Nusra Front	0.00	0.00	0.00	1
Armed opposition groups	0.60	0.25	0.35	12
International coalition forces	0.70	0.64	0.67	11
Other	0.75	0.72	0.73	61
Russian troops	0.53	0.53	0.53	15
Self administration forces	0.50	0.33	0.40	3
Syrian government and affiliated militias	0.60	0.61	0.60	46
The organization of the Islamic State in Iraq and the Levant - ISIS	0.33	0.54	0.41	13
Average/Total	0.63	0.60	0.61	162

Table 7.38: Classification report of Logistic Regression Actor classifier on FA-KES Titles and Contents testing dataset with tags predicted by OpenTag

### 7.3.2 Location Classifier

We implemented three classifiers for attribute location: Logistic Regression, LinearSVC, and Random Forest Classifier. We will see in the following subsections the results of the best performing one for each dimension.

### 7.3.2.1 FA-KES Titles Dataset

LinearSVC gave the best results among the three location classifiers. Tables 7.39 and 7.40 show the classification report of our classifier on the articles' titles (excluding tags) and on the articles' titles with the labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Aleppo	0.73	0.81	0.77	43
Damascus	0.75	0.86	0.80	7
Damascus Suburbs	0.62	0.62	0.62	16
Daraa	0.40	0.40	0.40	5
Deir Ezzor	0.50	0.71	0.59	7
Hama	0.25	0.50	0.33	4
Hasakeh	0.17	0.50	0.25	4
Homs	0.67	0.67	0.67	9
Idlib	0.61	0.69	0.65	16
Lattakia	0.60	1.00	0.75	3
Other	0.71	0.15	0.25	33
Quneitra	0.00	0.00	0.00	1
Raqqa	0.62	0.71	0.67	14
Average/Total	0.64	0.60	0.58	162

Table 7.39: Classification report of the baseline LinearSVC Location classifier on FA-KES Titles testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Aleppo	0.80	0.81	0.80	43
Damascus	0.75	0.86	0.80	7
Damascus Suburbs	0.71	0.62	0.67	16
Daraa	0.40	0.40	0.40	5
Deir Ezzor	0.45	0.71	0.56	7
Hama	0.29	0.50	0.36	4
Hasakeh	0.22	0.50	0.31	4
Homs	0.43	0.67	0.52	9
Idlib	0.58	0.69	0.63	16
Lattakia	0.60	1.00	0.75	3
Other	0.67	0.24	0.36	33
Quneitra	0.00	0.00	0.00	1
Raqqa	0.64	0.64	0.64	14
Average/Total	0.64	0.61	0.60	162

Table 7.40: Classification report of LinearSVC Location classifier on FA-KES Titles testing dataset with tags predicted by OpenTag

### 7.3.2.2 FA-KES Titles and First Paragraphs Dataset

In this case, Logistic Regression also performed better than the other location classifiers. Tables 7.41 and 7.42 show the classification report of our classifier

on the articles' titles and first paragraphs (excluding tags) and on the articles' titles concatenated with first paragraphs and their labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Aleppo	0.83	0.97	0.90	66
Damascus	0.67	0.50	0.57	4
Damascus Suburbs	0.53	0.80	0.64	10
Daraa	0.67	0.67	0.67	3
Deir Ezzor	0.81	0.76	0.79	17
Hama	1.00	0.60	0.75	5
Hasakeh	1.00	0.56	0.71	9
Homs	1.00	0.88	0.93	8
Idlib	0.92	0.71	0.80	17
Lattakia	1.00	0.50	0.67	4
Other	0.50	0.50	0.50	2
Quneitra	0.00	0.00	0.00	3
Raqqa	0.81	0.93	0.87	14
Average/Total	0.82	0.81	0.80	162

Table 7.41: Classification report of the baseline Logistic Regression Location classifier on FA-KES Titles and First Paragraphs testing dataset



<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Aleppo	0.83	0.97	0.90	66
Damascus	0.67	0.50	0.57	4
Damascus Suburbs	0.53	0.80	0.64	10
Daraa	0.67	0.67	0.67	3
Deir Ezzor	0.81	0.76	0.79	17
Hama	1.00	0.60	0.75	5
Hasakeh	0.83	0.56	0.67	9
Homs	1.00	0.75	0.86	8
Idlib	0.86	0.71	0.77	17
Lattakia	1.00	0.50	0.67	4
Other	0.50	0.50	0.50	2
Quneitra	1.00	0.33	0.50	3
Raqqa	0.93	0.93	0.93	14
Average/Total	0.83	0.81	0.81	162

Table 7.42: Classification report of Logistic Regression Location classifier on FA-KES Titles and First Paragraphs testing dataset with tags predicted by OpenTag

### 7.3.2.3 FA-KES Titles and Contents Dataset

Logistic Regression performed better than the other location classifiers. Tables 7.43 and 7.44 show the classification report of our classifier on the articles' titles

and contents (excluding tags) and on the articles' titles concatenated with their content and the labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Aleppo	0.58	0.72	0.64	53
Damascus	0.25	0.20	0.22	5
Damascus Suburbs	0.62	0.62	0.62	8
Daraa	0.33	0.25	0.29	4
Deir Ezzor	0.47	0.64	0.54	11
Hama	1.00	0.25	0.40	4
Hasakeh	0.40	0.44	0.42	9
Homs	0.50	0.25	0.33	8
Idlib	0.80	0.52	0.63	23
Lattakia	0.00	0.00	0.00	2
Other	0.46	0.44	0.45	25
Quneitra	0.00	0.00	0.00	1
Raqqqa	0.44	0.44	0.44	9
Average/Total	0.55	0.53	0.52	162

Table 7.43: Classification report of the baseline Logistic Regression Location classifier on FA-KES Titles and Contents testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Aleppo	0.59	0.68	0.63	53
Damascus	0.25	0.20	0.22	5
Damascus Suburbs	0.50	0.88	0.64	8
Daraa	0.33	0.25	0.29	4
Deir Ezzor	0.55	0.55	0.55	11
Hama	1.00	0.25	0.40	4
Hasakeh	0.36	0.44	0.40	9
Homs	0.50	0.12	0.20	8
Idlib	0.88	0.61	0.72	23
Lattakia	0.00	0.00	0.00	2
Other	0.44	0.44	0.44	25
Quneitra	0.00	0.00	0.00	1
Raqqa	0.33	0.44	0.38	9
Average/Total	0.55	0.53	0.52	162

Table 7.44: Classification report of Logistic Regression Location classifier on FA-KES Titles and Contents testing dataset with tags predicted by OpenTag

### 7.3.3 Cause of Death Classifier

We implemented three classifiers for attribute cause of death: Logistic Regression, LinearSVC, and Random Forest Classifier. We will see in the following

subsections the results of the best performing one for each dimension.

### 7.3.3.1 FA-KES Titles Dataset

LinearSVC gave the best results among the three classifiers of the cause of death. Tables 7.45 and 7.46 show the classification report of our classifier on the articles' titles (excluding tags) and on the articles' titles with the labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Chemical and toxic gases	0.67	0.83	0.74	12
Execution	1.00	0.80	0.89	5
Explosion	0.74	0.88	0.80	16
Other	0.67	0.35	0.46	52
Shelling	0.71	0.86	0.77	70
Shooting	0.27	0.50	0.35	6
Average/Total	0.69	0.68	0.66	161

Table 7.45: Classification report of the baseline LinearSVC Cause of Death classifier on FA-KES Titles testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Chemical and toxic gases	0.65	0.92	0.76	12
Execution	1.00	0.80	0.89	5
Explosion	0.70	0.88	0.78	16
Other	0.68	0.33	0.44	52
Shelling	0.71	0.86	0.78	70
Shooting	0.27	0.50	0.35	6
Average/Total	0.69	0.68	0.66	161

Table 7.46: Classification report of LinearSVC Cause of Death classifier on FA-KES Titles testing dataset with tags predicted by OpenTag

### 7.3.3.2 FA-KES Titles and First Paragraphs Dataset

In this case, LinearSVC also performed better than the other cause of death classifiers. Tables 7.47 and 7.48 show the classification report of our classifier on the articles' titles and first paragraphs (excluding tags) and on the articles' titles concatenated with first paragraphs and their labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Chemical and toxic gases	0.67	0.17	0.27	12
Execution	0.67	0.29	0.40	7
Explosion	0.40	0.09	0.15	22
Other	0.62	0.26	0.37	19
Shelling	0.63	0.96	0.76	93
Shooting	0.00	0.00	0.00	9
Average/Total	0.57	0.62	0.54	162

Table 7.47: Classification report of the baseline LinearSVC Cause of Death classifier on FA-KES Titles and First Paragraphs testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Chemical and toxic gases	0.67	0.17	0.27	12
Execution	1.00	0.29	0.44	7
Explosion	0.70	0.32	0.44	22
Other	0.47	0.37	0.41	19
Shelling	0.65	0.91	0.76	93
Shooting	0.50	0.11	0.18	9
Average/Total	0.65	0.64	0.59	162

Table 7.48: Classification report of LinearSVC Cause of Death classifier on FA-KES Titles and First Paragraphs testing dataset with tags predicted by OpenTag

### 7.3.3.3 FA-KES Titles and Contents Dataset

LinearSVC performed better than the other cause of death classifiers. Tables 7.49 and 7.50 show the classification report of our classifier on the articles' titles and contents (excluding tags) and on the articles' titles concatenated with their content and the labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Chemical and toxic gases	0.80	0.50	0.62	8
Execution	0.57	0.80	0.67	5
Explosion	0.84	0.80	0.82	20
Other	0.64	0.53	0.58	40
Shelling	0.68	0.83	0.75	78
Shooting	0.67	0.18	0.29	11
Average/Total	0.69	0.69	0.68	162

Table 7.49: Classification report of the baseline LinearSVC Cause of Death classifier on FA-KES Titles and Contents testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Chemical and toxic gases	0.80	0.50	0.62	8
Execution	0.50	0.80	0.62	5
Explosion	0.94	0.80	0.86	20
Other	0.65	0.60	0.62	40
Shelling	0.71	0.83	0.76	78
Shooting	0.67	0.18	0.29	11
Average/Total	0.72	0.71	0.70	162

Table 7.50: Classification report of LinearSVC Cause of Death classifier on FA-KES Titles and Contents testing dataset with tags predicted by OpenTag

### 7.3.4 Civilians Classifier

We implemented three classifiers for attribute related to the number of dead civilians: Logistic Regression, LinearSVC, and Random Forest Classifier. We will see in the following subsections the results of the best performing one for each dimension.

#### 7.3.4.1 FA-KES Titles Dataset

Logistic Regression gave the best results among the three civilians classifiers. Tables 7.51 and 7.52 show the classification report of our classifier on the articles' titles (excluding tags) and on the articles' titles with the labels predicted by our



deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Greater than 100	1.00	1.00	1.00	1
Greater than 50	0.67	0.40	0.50	5
Less than 50	0.98	0.99	0.99	156
Average/Total	0.97	0.98	0.97	162

Table 7.51: Classification report of the baseline Logistic Regression Civilians classifier on FA-KES Titles testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Greater than 100	0.00	0.00	0.00	1
Greater than 50	1.00	0.40	0.57	5
Less than 50	0.97	1.00	0.99	156
Average/Total	0.97	0.98	0.97	162

Table 7.52: Classification report of Logistic Regression Civilians classifier on FA-KES Titles testing dataset with tags predicted by OpenTag

#### 7.3.4.2 FA-KES Titles and First Paragraphs Dataset

In this case, LinearSVC performed better than the other civilians' classifiers. Tables 7.53 and 7.54 show the classification report of our classifier on the articles' titles and first paragraphs (excluding tags) and on the articles' titles concatenated

with first paragraphs and their labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Greater than 100	0.67	0.57	0.62	7
Greater than 50	0.60	0.27	0.37	11
Less than 50	0.93	0.97	0.95	144
Average/Total	0.89	0.91	0.90	162

Table 7.53: Classification report of the baseline LinearSVC Civilians classifier on FA-KES Titles and First Paragraphs testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Greater than 100	0.67	0.57	0.62	7
Greater than 50	0.50	0.27	0.35	11
Less than 50	0.93	0.97	0.95	144
Average/Total	0.89	0.91	0.90	162

Table 7.54: Classification report of LinearSVC Civilians classifier on FA-KES Titles and First Paragraphs testing dataset with tags predicted by OpenTag

### 7.3.4.3 FA-KES Titles and Contents Dataset

Logistic Regression performed better than the other civilians' classifiers. Tables 7.55 and 7.56 show the classification report of our classifier on the articles' titles

and contents (excluding tags) and on the articles' titles concatenated with their content and the labels predicted by our deep learning model respectively.

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Greater than 100	0.50	0.29	0.36	14
Greater than 50	0.83	0.31	0.45	16
Less than 50	0.88	0.98	0.93	132
Average/Total	0.84	0.86	0.83	162

Table 7.55: Classification report of the baseline Logistic Regression Civilians classifier on FA-KES Titles and Contents testing dataset

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
Greater than 100	0.56	0.36	0.43	14
Greater than 50	1.00	0.31	0.48	16
Less than 50	0.88	0.98	0.93	132
Average/Total	0.86	0.86	0.84	162

Table 7.56: Classification report of Logistic Regression Civilians classifier on FA-KES Titles and Contents testing dataset with tags predicted by OpenTag

## 7.4 Discussion

Logistic regression outperformed Random Forest Classifier and LinearSVC classifiers for actor classification on the three FA-KES subsets: Titles, Titles and First

Paragraphs, Titles and Contents. For the FA-KES Titles and First Paragraphs subset, we could see in Table 7.36 that the actor classifier that was tested on the output of OpenTag model scored 0.77 for F1-score which is greater than the score 0.76 reached by the baseline actor classifier in Table 7.35. Table 7.36 shows the result of classifying the articles in FA-KES Titles and Paragraphs testing dataset taking into consideration the labels that were predicted for each word in this dataset by our OpenTag model. We could see that “Al-Nusra Front” actor wasn’t detected by the actor classifiers (F1-score is equal to 0) and this is due to having only 6 articles categorized as “Al-Nusra Front” in the training FA-KES Titles and First Paragraphs subset. Thus, it is expected to not have the actor classifiers recognize the ones linked to “Al-Nusra Front”. Additionally, our actor classifiers that were tested on the output of our deep learning model performed better than the baseline classifiers on Titles and Titles and Contents subsets. This could be seen in Table 7.34 where the average F1-score is 0.61 whereas it scored 0.60 by the baseline actor classifier examined on Titles testing dataset. Moreover, Table 7.38 shows an F1-score average of 0.61 on Titles and Contents testing dataset though it scored 0.59 by the baseline actor classifier on the same testing data excluding tags.

Here comes the Location Classifier’s turn. We noticed that in this case LinearSVC performed better than Random Forest Classifier and Logistic Regression classifiers when tested on Titles subset. Tables 7.39 and 7.40 show 0.58 and 0.60 for F1-score respectively, so for the Titles subset, the LinearSVC location classifier

outperformed the baseline classifier. For the remaining FA-KES subsets, Logistic Regression performed better than Random Forest Classifier and LinearSVC classifiers. Tables 7.41 and 7.42 return 0.80 and 0.81 respectively as F1-score for the Titles and First Paragraphs testing subsets. For Quneitra classification, the location classifier that was trained on FA-KES Titles and First Paragraphs along with the ground-truth tags, scored 0.50 for F1-score when tested on the output of OpenTag model whereas the baseline Location classifier didn't detect this class (F1-score is equal to 0). This highlights the importance of tags that helped in this case to get a more accurate result. Moreover, we could see that Tables 7.43 and 7.44 show the same F1-score on average 0.52 which means that both Logistic Regression location classifiers that were trained on FA-KES Titles and Contents with and without tags performed equally on the testing dataset.

LinearSVC classifiers for the cause of death performed the best among all the other classifiers for all FA-KES subsets. We could see that the highest result is scored for the Titles and Contents testing dataset where F1-score reached 0.70 (Table 7.50), whereas it got 0.68 by the baseline cause of death classifier (Table 7.45). If we check the results of the other cause of death classifiers, we could see in Tables 7.45 and 7.46 that both classifiers performed on average the same where F1-score reached 0.66. Also, the cause of death classifier that was trained on FA-KES Titles and First Paragraphs along with ground-truth labels, scored 0.59 as F1-score (see Table 7.48) when tested on the output of our recurrent neural network model, whereas the baseline classifier reached 0.54 for F1-score

(see Table 7.47). We noticed that the “Shooting” class wasn’t detected by our baseline classifier when tested on Titles and First Paragraphs testing subset since it scored 0 for F1-score as is shown in Table 7.47. However, the cause of death classifier that was tested on the output of the OpenTag model gave 0.18 as F1-score for the “Shooting” class which highlights the importance of tags in this case.

Lastly, Logistic regression classifiers for categorizing the number of dead civilians got the highest F1-score which is 0.97 when examined on the Titles testing dataset as can be seen in Tables 7.51 and 7.52. Now, for the Titles and First Paragraphs testing subset, Tables 7.53 and 7.54 show the same F1-score on average for both LinearSVC classifiers which is 0.90. Finally, Logistic regression civilians classifier that was tested on the output of our OpenTag model for the Titles and Contents testing dataset outperformed the baseline classifier by scoring 0.84 for F1-score as is shown in Table 7.56 whereas the baseline classifier reached 0.83 for F1-score as can be seen in Table 7.55. Civilians classifiers gave very good results whether they were initially trained on the text only or with additional features.

# Chapter 8

## Conclusion

In this study, we worked on incident extraction from crisis data that report interesting information such as the number of casualties and location of the incident. To reach this goal, we implemented OpenTag, a deep sequence-tagging approach proposed in [1]. We trained our model on the training dataset of each subset of Joplin and FA-KES datasets. Then, we evaluated our paradigm on the corresponding testing dataset of Joplin and FA-KES subsets. Additionally, we created a baseline model, trained it on the same learning datasets, and tested it on the appropriate testing datasets to see how well our model is doing compared to a baseline model. Moreover, for the FA-KES dataset, we created classifiers for the following attributes: Actor, Civilians, Location, and Cause of death. Furthermore, we implemented baseline classifiers and tested them on the testing FA-KES subsets.

Our OpenTag model outperformed the baseline model CRF for the Caution

and Advice and Information Source subsets of the Joplin dataset. However, the baseline CRF model returned better results than OpenTag when evaluated on the testing datasets of the FA-KES subsets. As for the classifiers that were trained on the FA-KES subsets along with ground-truth tags, they were evaluated on the output of our OpenTag model. These classifiers either performed equally or better than the baseline classifiers returning the following results: actor classifier scored 0.77 as F1-score when tested on FA-KES Titles and First Paragraphs subset, location classifier scored 0.81 as the highest value for F1-score for FA-KES Titles and First Paragraphs testing dataset, cause of death classifier scored the highest F1-score which is 0.70 for the FA-KES Titles and Contents testing subset, and lastly, civilians classifier recorded 0.97 for F1-score when assessed on the FA-KES Titles testing subset.



# Appendix A

## Abbreviations

NER	Named Entity Recognition
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
CRF	Conditional Random Field
NLP	Natural Language Processing
RNN	Recurrent Neural Network
CIV	Civilians
NCV	Non-civilians
CHD	Children
DAT	Date
LOC	Location
COD	Cause of death

PEO	People
INF	Infrastructure
TIM	Time
INC	Incident
DON	Donation
REP	Authority responsible
SRC	Information source
WMN	Women

# Bibliography

- [1] G. Zheng, S. Mukherjee, X. L. Dong, and F. Li, “Opentag: Open attribute value extraction from product profiles,” *CoRR*, vol. abs/1806.01264, 2018.
- [2] W. Koehrsen, “Beyond accuracy: Precision and recall.” <https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c>, Mar 2018.
- [3] “The essential high-quality data annotation platform.” <https://www.figure-eight.com/>.
- [4] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, “Extracting information nuggets from disaster- related messages in social media,” 05 2013.
- [5] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, “Practical extraction of disaster-relevant information from social media,” 05 2013.
- [6] F. K. Abu Salem, R. Al Feel, S. Elbassuoni, M. Jaber, and M. Farah, “Fa-kes: A fake news dataset around the syrian war,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 13, pp. 573–582, Jul. 2019.
- [7] A. Salem, Fatima, A. Feel, Jaber, Mohamad, Farah, and Roaa, “Dataset for fake news and articles detection,” Jan 2019.
- [8] “Regularizers.” <https://keras.io/regularizers/>.
- [9] M. E. Korkmaz, S. Güney, and Ađule Yüksel YiÄşiter, “The importance of logistic regression implementations in the turkish livestock sector and logistic regression implementations/fields,” 2012.
- [10] A. Vora, “Classification – one vs rest and one vs one.” <https://datastoriesweb.wordpress.com/2017/06/11/classification-one-vs-rest-and-one-vs-one/>, Jul 2019.
- [11] S. Patel, “Chapter 5: Random forest classifier.” <https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1>, May 2017.

- [12] “Linear svc machine learning svm example with python.” <https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/>.
- [13] A. Mishra and A. Mishra, “Metrics to evaluate your machine learning algorithm.” <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>, Feb 2018.
- [14] “Classification: Accuracy | machine learning crash course | google developers.” <https://developers.google.com/machine-learning/crash-course/classification/accuracy>.
- [15] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” 10 2002.
- [16] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O’Reilly Media Inc.