Mining Large-scale News Articles for Predicting Forced Migration via Machine Learning Techniques

Forouq Khonsari

a thesis submitted to
the faculty of graduate studies
in partial fulfilment of the requirements
for the degree of
Master of Science

graduate program in Electrical Engineering and Computer Science
York University
Toronto, Ontorio

May, 2018

Abstract

Many people are being displaced from their homes every day from all around the globe. An estimated 12.4 million people are displaced due to conflict or persecution in recent years. Many of them are forced to leave their homes because of socio-political conflicts, human-made or natural disasters.

In order to develop an early warning system for forced migration in the context of humanitarian crisis, it is essential to study the factors that cause forced migration, and build a model to predict the future number of displaced people. In this research, we focus on studying forced migration due to socio-political conflicts for which violence is the main reason. In particular, we investigate whether the degree of violence in a specific region can be detected from news articles related to that region and whether the detected violence scores can be used to improve the prediction accuracy.

Furthermore, any incident happening within the environment of concern could be a possible trigger for forced migration. As a result, in order to thoroughly investigate all the possible triggers, we need to rely on a complete source to capture useful information about the factors of forced migration. According to our needs, news articles seem to be a powerful resource for our research.

Our proposed framework uses a large corpus of news articles to extract the factors and signals of forced migration from. In particular, we pay attention to violence as the main factor of forced migration, however, we also attempt to extract other useful features from news documents including emotions. Our framework uses three different techniques for measuring the degree of violence for a specific environment of concern. The first technique which is called *SWSW*, measures the semantic similarity between documents and a set of seed-words representing violence. The second technique, also called *ED-FE*, aims at extracting violent events from news articles. In particular, these violent events are incidents related to attacks or the ones resulting in casualties. The third technique is called *TD-FE* and it intends to process the corpus of news articles by using topic modeling techniques to reduce the size of the information and represent the whole corpus in

a more efficient and compact format to further analyze and filter the information related to violence in the region. Also, emotions are extracted using a selected keyword-based technique which outputs 6 basic emotions detected from news articles, such as *Anger* and *Disgust*. At last, in the last stage of our framework, the extracted violence and emotion scores are gathered together to build a powerful feature set for forced displacement prediction models.

After all, extensive experiments were conducted to evaluate the prediction models using real world datasets. Experiments indicate that ED-FE and TD-FE provide accurate violence scores which are very effective features for making forced displacement forecasts. Moreover, Anger and Disgust scores also proved to be effective in predicting forced migration, and using all these features in prediction models has shown to improve the prediction accuracy.

Contents

List of Tables

Listing of figures

Migration is an expression of the human aspiration for dignity, safety and a better future. It is part of the social fabric, part of our very make-up as a human family.

Ban Ki-moon

1 Introduction

1.1 Motivation

Many people from all around the world have been forced to leave their homes due to conflict or persecution in recent years. According to United Nations high commission for refugees, the number of displaced population has soared reaching 65.3 million, [1].

A state of forced migration exists when significant number of people in a given locality have been displaced by socio-political conflicts, human-made or natural disasters, economic disturbance, disease or development projects, [2]. In order to develop an early warning system for forced migration in the context of humanitarian crisis, it is essential to study the factors that cause forced migration, and build a model to predict the future number of displaced people. Addressing the causes of movements and preparing for emergency movements are some of the benefits of early warnings for mass displacement.

Thus, in this research we study the factors of forced migration and we are particularly focusing on Iraq as a case study. Social science experts have studied forced migration in Iraq and have distinguished some factors influencing forced displacement, referred to as social-factors:

- Violence and non-safety in the country for which violent extremist groups are the main reason. People without protection would experience persecution and other serious human right violations which force them to migrate. Social scientists who have studied forced migration in Iraq have identified violence and security threat in the region to be the most important factor influencing forced displacement, [3].

- Relief or help received from other countries or organizations helps many people survive from violence or starvation and thus gives them reasons to stay.

- Political interference and lack of effective government including govern-

ment's incapability to defeat extremists, as well as religious, ethnic and political differences including conflicts between Sunni and Shia people, forces people to leave the country.

- Economic issues including high unemployment and inflation rate make people leave their homes in search of jobs, [4].

Furthermore, people are affected by their environment, other people and incidents happening around them. These factors can make someone decide whether to leave their home. Thus, detecting and identifying the events * happening inside the environment of concern is important for studying forced migration. News agencies produce a vast amount of information about the events happening every day, covering from local to international affairs. The content of news articles ranges from social, political, economic developments, to reports on the environmental events. As a result, the news articles collected from across the globe are a suitable source for analyzing the world events.

The objective of this research is to use machine learning and natural language processing techniques to develop a model for predicting the number of displaced people (e.g. refugees and asylum seekers) based on big data of news articles. This research is particularly focused on forced migration due to socio-political conflicts for which violence and non-safety is the main element.

*An event is defined as an incident happening in a specific location involving entities, time and location

1.2 Proposed Framework

To solve the problem of how to build a prediction model for forecasting the future refugee movements, we introduce a framework to detect factors of forced population displacement by analyzing a corpus of news articles which is an excellent source to extract the latest events in the region. Our framework consists of four components. The first component includes filtering a huge corpus of news articles according to relativity to the location of interest. This is because the factors of forced population displacement differ with respect to the location.

The second component includes analyzing the filtered news articles to extract violence scores, which represent the degree of violence in a period of time at a location. Three techniques are proposed in this component to detect and measure violence from news articles, which are briefly described below.

- The first technique is called Similarity With Seed Words (SWSW) which measures the semantic similarity of the words of documents with a predefined set of seed words to assign a degree of violence to the documents. This information is further used to define a degree for violence.

- The second techniques is called Event Detection-Factor Extraction (ED-FE) which extracts the incidents containing killing, attacking or injuring from news articles and uses this information to assign a degree of violence on a periodical basis.

- The third technique is called Topic Detection based Factor Extraction (TD-FE) which extracts topics from the corpus of news articles and labels them

according to their relevance to violence. The labeled topics that occur in the articles in a period of time are then used to assign a degree of violence for that period of time.

The third component extracts six emotions from news articles, including anger, disgust, fear, happiness, sadness and surprise. The reason to do so, is that sentiments of news give us a good understanding of their possible impacts on people. Events happening around people might act as a trigger to change one's mind towards migration and thus studying the sentiments of news is important in the context of predicting forced displacement. A rise in the number of fearful or sad news could be a good signal for forced migration, which could be used as an early alarm for a rise in the number of refugees.

At last, in the forth component, we apply machine learning techniques to develop prediction models and investigate whether the violence and emotion scores extracted from news articles are helpful in predicting forced migration. A number of regression models are used in our application, such as linear regression, neural networks, and random forest. We first build time-series autoregressive models using the above techniques and then add the extracted violence and emotion scores as input to the models. The extracted violence and emotion scores are evaluated by measuring the improvement observed by adding them to the prediction models.

Our experiments indicate that ED-FE and TD-FE violence scores as well as Anger and Disgust scores detected from news articles are effective factors for predicting forced migration, and together with lagged variables of time-series, they make a good feature set for building prediction models for forced migration.

1.3 Contributions and Challenges

The contributions of this thesis is summarized as follows:

- We propose novel methods for extracting degrees of factors for forced migration from news articles.
- We present a novel application of machine learning and natural language processing techniques to prediction of forced migration based on news articles.
- We incorporate the violence and emotion scores detected from news articles into the forced migration prediction model, and demonstrate through experiments that the detected scores are effective in predicting forced migration.
- Human-generated databases are created for further evaluation and comparison of the factors of forced migration.

The overall challenges of this research include:

- The quality of the factors extracted by our proposed framework fairly depends on the level they are covered by news agencies. The coverage quality of the news articles determines the accuracy and soundness of the extracted factors. Thus, it is hugely important in this research to use a complete and through corpus of news articles, covering all the news about the location of interest and for all the dates during the time-period of our focus.

- Lack of related research about extracting factors of forced migration from text, or building prediction models for forced migration using these factors, leaves us with no ground truth or predefined baseline model for evaluation of the extracted factors or prediction models. Thus, human judgment had to be used in some steps of the framework for evaluation and further analysis. Also, baseline models need to be created for evaluating the final prediction models.

1.4 Thesis Outline

The rest of the thesis is organized as follows. In chapter 2 we discuss the related work and since our framework consists of several components, each section of related work is related to a separate component. After that, we explain the structure of our framework in chapter 3 and describe each component in detail. At last, the experimental settings and datasets are explained in chapter 4 and the final results about applying our framework on real world datasets are reported in chapter 5.

Research is to see what everybody else has seen,
and to think what nobody else has thought.

Albert Szent-Gyorgyi

2

Related Work

This research covers various domains in text processing, and it also lays on the intersection of social sciences and computer science. Thus, we have explored different domains of text processing in order to extract useful features for our prediction models from news articles. The main text processing techniques we have considered in this research include event detection, emotion detection and topic

detection from text. We have distinguished the related work to this matter in four sections. First we discuss previous research on studying and predicting forced migration. Then we describe related work regarding event and topic detection from text. And at last, an overview of the emotion detection techniques is provided.

2.1 Forced Displacement Prediction

[5] have previously discussed the problems associated with indicator analysis for the purpose of early warning for forced displacement. They argue that the most trivial problem is the matter of timing of indicators. They state that exact timing of indicators may never be possible according to the fact that each incident of forced migration has particular characteristics. In other words, root causes (or long-term causes) of forced displacement may occur years before the displacement actually takes places, while medium-term (or proximate) causes could happen only a few months before the incident of forced migration. The authors state that "triggering events" are the most difficult to discover and analyze. Theoretically, triggering events take place only a few days before the mass displacement. Furthermore, most conventional methods, including time-series analysis, lack the ability to take into account the close timing associated with triggering events. According to the aforementioned problems, researchers mostly ignore immediate causes such as triggering events, and mainly concentrate on long-term factors. However, it is important to keep in mind that the immediate causes are most critical for decision makers in terms of preparing for emergency relief.

In 1997, [6] developed a theoretical model of refugee migration based on the fac-

tors with estimated magnitude. Some of these factors include economical underdevelopment, human rights violation, ethnic and civil conflicts. These factors were then included in a pooled time-series analysis to predict the number of refugees. This research showed that economic and intervening policy variables are less useful for predicting refugee migration than the threat of violence. This work is different from ours regarding the methodology used for extracting the forced migration signals. Unlike this research that uses manually generated scores from various resources for the factors mentioned above, we automatically extract the scores of the factors of forced migration from news articles.

To the best of our knowledge, there is no previous work on predicting forced migration using news articles or any other kind of text. Factor extraction from text, on the other hand, has been investigated by [7]. They introduced a method to extract the magnitude of violence from news articles. This method uses word embedding techniques to embed the words of news articles and then uses similarity measures within the embedding space to compute the similarity between the words of a document and a set of predefined seed words indicating violence. At last, a correlation was observed between the extracted violence scores and the number of migrated people. This work only detects the magnitude of violence from news articles and does not focus on other factors of forced migration. The quality of the extracted violence scores considerably depends on the quality of the manually generated set of seed words.

2.2 Event Detection

Fortunately, there has been a lot of research in event detection and extraction from text in the past ten years. The primary event detection methods use predefined (or learned) patterns to identify event triggers (event trigger is a phrase, most often a single verb or noun, associated with each event mention, which evokes that event.) and event arguments among sentences. We can make a rough distinction between two types of patterns that can be applied to natural language corpora for event extraction, i.e., lexico-syntactic patterns, [8], and lexico-semantic patterns, [9]. The former patterns are a combination of lexical representations and syntactic information. The latter patterns are more expressive, and combine lexical representations with both syntactic and semantic information. Lexico-syntactic patterns often appear in earlier work on knowledge-driven event extraction [10], but have remained popular in more recent approaches [8] [11] due to their domain independence. The patterns mostly rely on syntactic properties (grammatical meanings) like verbs, nouns, prepositions, and pronouns.

Later studies in event extraction include using classifiers to detect trigger words in sentences, [12, 13, 14, 15, 16, 17, 18]. These methods include hand-designing a large set of features to be fed into the statistical classifiers. Although this approach has achieved a high performance, [19, 20], it suffers from at least two issues. First, the choice of features is a manual process and includes much effort and time. Second, the NLP toolkits used for feature extraction (i.e name taggers, parsers, semantic role labelers, gazetteers etc) might have errors which propagate

through other phases of event extraction.

Advanced studies in event extraction during the past three years rely on using deep neural network models for event extraction tasks. [21] were the first to use Convolutional Neural Networks to detect the event triggers and event types in sentences. This model lacks the ability to model the consecutive k-grams and ignores the non-consecutive k-grams that might involve important structures for event detection, thus, non-consecutive convolution was introduced which is capable of capturing long term dependencies between words in a sentence, [22]. According to the fact that events and entities are closely related, entities are often actors or participants in events and events without entities are uncommon. The method introduced by [23], models the dependencies among variables of events, entities, and their relations, and performs joint inference of those variables across a document. All the previous works perform best on single-event sentences while a great portion of the sentences have more than one event (Multi-event sentences). Therefore, Dynamic Multi-Pooling Convolutional Neural Networks were introduced to overcome this limitation by using a dynamic multi-pooling layer according to event triggers and arguments, to reserve more crucial information, [24].

Some researchers have been focusing on extracting events as a set of related and semantically similar sentences reported from different sources of information. A graph based algorithm was proposed by [25] to detect the events and story lines about a target domain from a massive set of news articles. This work produces only the summary of events as the output and does not focus on detecting the magnitude or score for the events.

2.3 Topic Detection

There are different approaches to deal with documents. Extracting syntactic or semantic structures such as grammatical patterns or entity roles are some examples of the extractable information from text. To discover and extract themes from documents, topic modeling algorithms are used. The main goal of topic modeling algorithms is analyzing the semantic layer of huge datasets of documents.

Fortunately, we have witnessed a lot of improvement in the domain of topic modeling during the past years. Latent Dirichlet Allocation (LDA), [26], might be the most common-known probabilistic topic modeling algorithm. It defines a topic as a distribution over the vocabulary and it is based on the assumption that each document is a distribution over topics. LDA is applied based on mixture models and uses Dirichlet distribution as its prior of some parameters. LDA is a generative model which assumes that documents are created from a generative process. This generative process contains hidden variables derived from a distribution. A joint distribution is defined over all observed words (in the documents) and latent variables (topic distributions). One of the main limitations of LDA is its inability to model topic correlation. In other words, LDA is not a powerful tool for learning the correlation between various documents with similar topics.

Correlated topic model (CTM), [27], is an extension of LDA with more focus on topic correlation modeling. CTM is created based on LDA except that unlike LDA in which topic proportion is derived from a Dirichlet distribution, CTM assumes that topic proportion comes from logistic normal distribution. One main

limitation of CTM is that it only captures the pairwise correlation between topics and lacks the ability to model correlations between multiple topics.

To overcome the aforementioned limitation, Pachinko Allocation Model (PAM) was introduced by [28]. PAM captures the topic correlations by directed acyclic graph (DAG). DAG was used to represent individual words in the vocabulary while each node of the graph models a correlation among its children. PAM is different from previously mentioned topic modeling algorithms such that it has expanded the definition of topic to the distribution over other topics as well as just words.

The main drawback of LDA and all the aforementioned models which were extensions of LDA, is that the structure of the document including the order of the words, is completely ignored. This is called bag-of-words assumption which does not take into account the sequence of the words. To solve this problem, Bi-gram topic model was introduced by [29], which combines LDA with bi-gram language model. Another approach is called Unsupervised Topic Segmentation (NTseg), introduced by [30], which is an unsupervised topic segmentation approach created to take into account the order of the words.

2.4 Emotion Detection

Algorithms for emotion detection from text are basically divided into two main categories: keyword-based techniques and learning-based techniques, [31]. Keyword-based techniques generally search for occurrences of some specific keywords within a piece of text, [32]. Usually there are constant predefined set of emotion cate-

gories used for this kind of research, such as disgusted, sad, happy, angry, fearful, surprised etc. Works regarding keyword-based emotion detection generally cover all the aforementioned categories of emotions. Keyword-based emotion detection algorithms generally include receiving a piece of text as input, transforming the input text into tokens and then detecting the emotion words from the tokens. The detected emotion words are then analyzed in terms of intensity and after all a final emotion category is identified as the output of the algorithm, [33]. Basically, the performance of keyword-based methods depends on two things: the quality of the predefined emotion keywords and the procedure of processing the input text to extract keywords, [34]. In general, keyword-based emotion detection algorithms have three main limitations, [35]:

1. It is hard to come up with a good set of keywords representing emotion classes while it is so challenging to decide how far to expand the dictionary, as most of the times the meaning of the words are determined within the context. The meaning of keywords might transform relatively to the surrounding words and even with a good dictionary of key-words representing emotion classes, it is still very challenging to distinguish cases of sarcasm or irony.

2. Selecting an emotion category for a sentence depends on the existence of a keyword in the sentence. So basically the sentences with no keyword are assumed to have no sentiment at all, which might not be the case in real world.

3. Keyword-based algorithms do not take into account the linguistic information of the sentence, such as syntax, which is helpful in some cases to decide on the emotion of the sentence.

The second category of emotion detection techniques are learning-based algorithms. Learning-based algorithms rely on a set of sentences labeled with predefined classes of emotions to train a classifier based on this dataset. Many different machine learning classification techniques can be used in this matter such as Bayesian classifiers, [36], support vector machines, conditional random fields or recurrent neural networks. In these algorithms, sentences are embedded and mapped into numbers so that the classifiers can accept the numbers as inputs. Depending on the embedding techniques, some embedding techniques are to some level capable of capturing syntax and semantics of the sentences and so they could embed all the necessary linguistic information of the sentences into vectors. Although this looks to be a stunning advantage over keyword-based techniques, learning-based algorithms are exposed to noise and outliers and their performance hugely depends on the quality of the labeled data such as correctness of the labels and not including missing data. Furthermore, to avoid over-fitting or under-fitting in these algorithms, an adequate amount of training data should be provided to guaranty high performance.

On the other hand, the most important advantage of key-word based emotion detection over learning-based algorithms is that the procedure is unsupervised and there is no need for labeled data. This comes in handy when access to labeled data is impossible or expensive.

Methodology is intuition reconstructed in tranquility.

Paul Lazarsfeld

3

Methodology

In this section we provide the details of our proposed framework consisting of four components. Figure 3.1 shows the overview of our proposed framework. The first component is *location detection* which extracts the news articles relevant to the location of interest (which is Iraq in this research). The second component is *factor extraction* which applies three techniques on a corpus of news articles to

extract and measure factors of forced migration. In this research we are focusing on violence as the most significant factor. The third component is *emotion detection* which extracts six categories of emotions out of news articles. The forth component is *building prediction models* using the emotions and violence scores extracted in previous components. At last, the extracted factors and emotions are evaluated in terms of how much they have improved the prediction results.

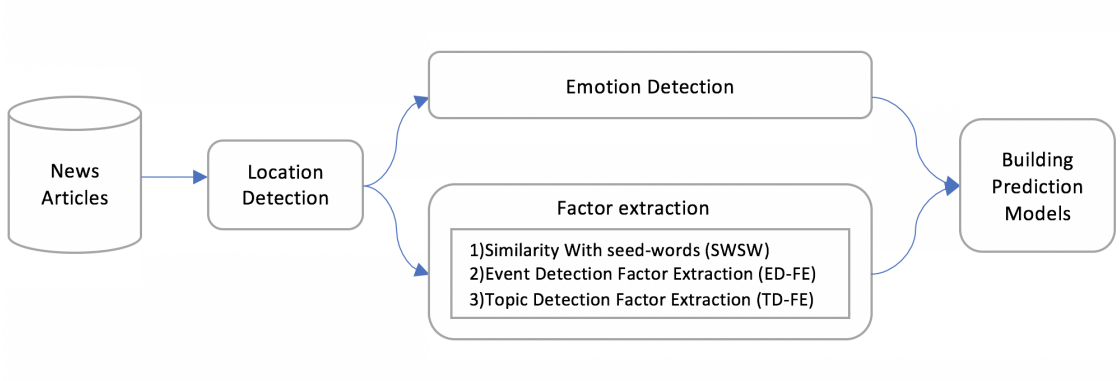

Figure 3.1: Our proposed framework and its components.

Below are the details of every component of our framework.

3.1 Location Detection of News Articles

We seek to find the location of each article in order to filter them based on relevance to Iraq. Location detection is the task of finding the location of the article. More specifically, the place where the article is talking about or where the main event of the article takes place. According to [37], words referring to place names may

- Occur as person names, e.g., 'Dair' is a girl name which is also name of a

city in Basra province of Iraq.

- Occur as common words, e.g., 'And' is a village in Iran, or 'Hit' is a city in Iraq.
- Have variants, e.g., well known cities have language variants, e.g., 'Seul' meaning alone/only in French is also the Capital of South Korea in Portuguese and Italian.
- Refer to different locations, e.g. both Baghdad and Kirkuk provinces of Iraq have a city called 'Al Rashid'.

We tested two main approaches for location detection which are described in next sub-sections. The first two of the mentioned problems were solved using Named Entity Recognition (NER) described in second sub-section and the third problem was resolved by filtering the non-English articles.

3.1.1 Location Detection by Frequency

In this approach, a gazetteer of place, city and provinces of Iraq is used to geo-match the list of words in the news articles. The most frequent location name is chosen as the location of the article. According to the fact the main event usually appears at the first paragraph of the article ([37]), we assume that location of the articles may also appear more likely at the beginning of the article. Thus, if two locations have the same frequency, the first one appearing in the article is chosen. The results were tested on 100 manually labeled articles and the accuracy was

71%. This approach is very simple and strict forward, however, doing this blindly runs into problems, as shown in Figure 3.2.

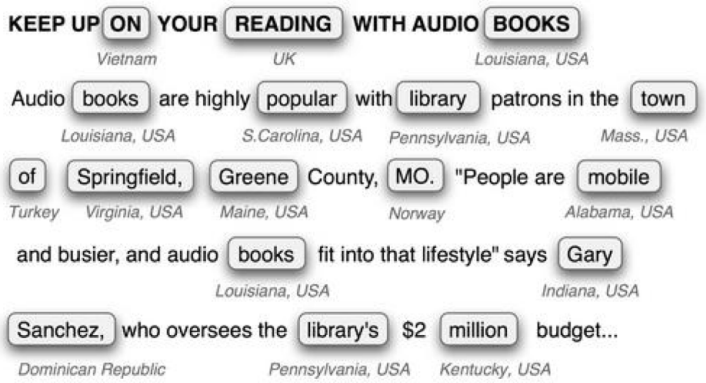

Figure 3.2: Some sample problems caused by blindly picking up the most frequent word as the location of the document.([38])

The gazetteer has good coverage of locations in Iraq and thus the location detection system incorrectly finds non-location words as places, such as mistakenly selecting the English word 'hit' as a city in Iraq. Of course we could omit such locations from the gazetteer, but then we would not be able to identify them when they do appear in the document. Thus, we used NER described in next section.

3.1.2 Location Detection Using Named Entity Recognition

Named entity recognition (NER) is a name for the task of locating and classifying named entities into predefined categories. The top categories for named entities are locations, organizations and names of people. One major source of difficulty is caused by the fact that many named entity terms are ambiguous. Thus May and North are likely to be parts of named entities for DATE and LOCATION, respectively, but could both be part of a PERSON; conversely Christian Dior

looks like a PERSON but is more likely to be of type ORGANIZATION. A term like Yankee will be ordinary modifier in some contexts, but will be marked as an entity of type ORGANIZATION in the phrase Yankee infielders ([38]).

Further challenges are posed by multi-word names like Stanford University, and by names that contain other names such as Cecil H. Green Library and Escondido Village Conference Service. NER is implemented by Bird et al, ([38]) and is published through NLTK* library. It consists of two main phases, detecting the named entities and classifying them. The definition of the term "named entity" is not strict and often has to be explained in the context in which it is used. NER begins by splitting the raw text of the document into sentences using sentence Segmenter, and each sentence is further subdivided into words using a Tokenizer. Next, each sentence is tagged with part-of-speech tags and then in the named entity detection phase, sentences are searched for mentions of potentially interesting entities using a classifier trained to recognize named entities. NER improves the task of location detection by capturing the names with more than one token such as: "Tel Afar", "Hammam Al Aleel" or "Tuz Khormato" (these are all names of cities in Iraq). Also, parts of speech other than names would not be considered as locations thus improving the precision. As an example, 'Hit' is a popular verb and noun in English, which was considered in our previous step as the city 'Hit' located in Anbar province. Therefore, when using NER, only the named entities categorized as locations are searched in the taxonomy to see if they belong to Iraq and the most frequent one is chosen. The accuracy for this

*http://www.nltk.org

method was 88% testing on the same corpus used for previous method.

3.2 Factor Extraction

3.2.1 Similarity With Seed Words (SWSW)

This technique was proposed by [7], in which a new method was introduced for detecting the magnitude of factors of forced migration from news articles. This method basically uses the word similarity measures to find the similarity between a document and a factor. This research focuses on a special factor in particular and relies on a set of manually generated seed words provided by social-scientists, which represent the predefined factor. Table3.1 shows examples of seed words related to some of the factors: violence, relief, economic instability and environmental threats.

Factor	Seed Words
Violence	violence, conflict, fight, killing, battle, massacre. butchery, injury, bombing, explosion, corpse, abduction, ambush, suicide, rape, persecution, assassination, terror, military, attack
Relief	Relief, disaster, emergency situation, refugee camp, tent, aid, host community, outbreak, infectious diseases, epidemic, disease, contagious, infection, donor, vaccination, campaign, reconstruction, supplies, medical, grant
Environmental Threats	natural resources, food scarcity, food shortage, drought, flood, environmental degradation, countryside, rural, agriculture, farmer, temperature, crop production, climate change

Table 3.1: Examples of some factors and their related seed words.

Similarity between a document and the set of seed words is considered as the average similarity of all its words with the set of seed words, and represents how relative each document is to the factor. Magnitude of the factor is then detected using the relevance of the documents to the factor over time, following three steps:

1. Relevance of a word to a factor (factor score for a word) is calculated by measuring the similarity between the mentioned word and every word in the set of seed words and finally taking the average of all the similarity values.

2. Relevance of a document to a factor (factor score for a document) is calculated by averaging over all the event scores of all the words in the document.

3. Factor score for a date is calculated by averaging over all the factor scores for all the documents in that day.

To measure the similarity between two words, two similarity measures were tested, explained in the following sections.

Normalized Point-Wise Mutual Information

Point-wise Mutual Information is a measure of shared information between two variables ([39]). The PMI between two random variables X and Y, is defined as: $I(X,Y)=log(p(x,y)/p(x)p(y))$. When values of X and Y have marginal probabilities $p(X)$ and $p(y)$, and joint probability $p(X.Y)$. In other words, mutual information compares the probability of x and y being seen together with the probability of seeing x and y independently. When X and Y are independent, which means $p(x,y)=p(x)p(y)$, is when the MI is 0. On the other hand, when x and y are completely correlated, MI obtains its highest value.

PMI was introduced into lexicography by [40]. Considering two words as the two variables of PMI, and measuring the mutual information for two words, comes from the idea that two words that occur closer together are semantically

more related and share more information with each other. PMI is not good for comparisons because it lacks a fixed upper bound. A normalization is performed to give PMI the value of 1 in case of the perfect correlation between the two variables. When the two words have perfect correlation it means that they occur only together. In other words, the chance of seeing x ($p(x)$) equals the chance of seeing y ($p(y)$) which is equal to the chance of seeing them together ((x,y)).

$$PMI(x,y) = -logp(x) = -logp(y) = -logp(x,y) \tag{3.1}$$

All above three options can be used for normalization, but the last one was chosen by ([39]) as it simplifies an upper bound as well as a lower band.

After all, Normalized Point-wise Mutual Information is defined as:

$$MPMI = \frac{log\frac{p(x,y)}{p(x)p(y)}}{-logp(x,y)} \tag{3.2}$$

NPMI is -1 when the two words never occur together and is 1 when they only occur together. When they are distributed as expected under independence, $NPMI(x,y) = 0$.

In this research, (x) and (y) are measured by counting the number of times x and y appear within a context window. (x,y) is measured by counting the number of times x and y occur together within a window. NPMI scores are calculated for every two words in our corpus and then they are used as a measure of similarity between the seed words and document words.

Cosine Similarity and Word Embedding

Word embedding is a technique where vectors of real numbers in a multidimensional continuous space are assigned to words or phrases from a vocabulary. Word vectors depend on the vocabulary size and are relative to the context. Word vectors are used to represent the meaning of the words which means the idea that is represented by the word or the idea that a person wants to express by using the word. Once the words are projected into a high dimensional vector space, the similarity between two words is measured using the cosine similarity between their vectors.

$$\text{Similarity}(word_1, word_2) =< word_1, word_2 >$$

There are different types of embedding techniques. The most popular technique is called *Word2Vec*. Mikolov et al, ([41], [42]) introduced two models, popularized by the Word2Vec[†] program, to generate these word vectors using neural networks, Skip-Gram (SG) and Continuous Bag-of-Words model (CBOW). Each of the aforementioned models use a two layer neural network which takes a text corpus as input and generates a set of feature vectors representing all the words in that corpus.

Continuous Bag of Words Model: The input of CBOW model could be $wi_{-2}, w_{i-1}, w_{i+1}, w_{i+2}$, the preceding and the following words of the specific

[†]https://code.google.com/archive/p/word2vec/

word that we want to generate the feature vector for. The output of the neural network is w_i. In other words, CBOW predicts the feature vector of a word using its context. The number of the preceding and following words used depends on the window size.

SKIP-GRAM MODEL: Unlike CBOW, the output of the skip-gram model could be $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$, and the input is w_i. Therefore, the task here is to predict a context using a word. Also, the context here is not limited to the immediate neighbors of the word and training instances can be created by skipping a constant number of words in its context, so for example, $w_{i-3,}w_{i-4}, w_{i+3}, w_{i+4}$. Figure 3.3 shows the architecture of the Skip-gram model.

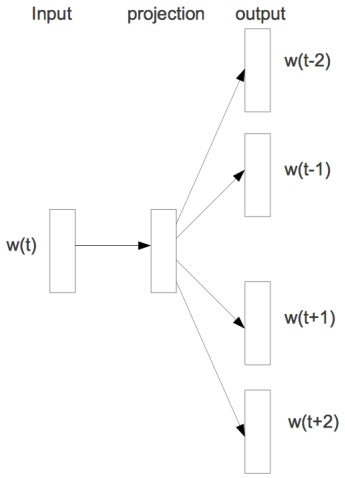

Figure 3.3: The Skip-gram model's architecture. The training objective is to learn word vector representations that are good at predicting the nearby words. ([42])

According to Mikolov: "Skip-Gram works well with small amount of the training data, represents well even rare words or phrases. CBOW is several times faster

to train than SG and has slightly better accuracy for the frequent words." Word vectors learned by CBOW and SG have a special characteristic which is semantically similar words occur in vicinity of each other and also these vectors are very good at encoding dimensions of similarity. Analogies testing dimensions of similarity can be solved quite well just by doing vector subtraction in the embedding space. Both syntactic and semantic similarities are captured in this way. The similarity between two words is calculated using cosine similarity of their vector representations within a fixed range of -1 to 1, similar to NPMI.

Following [7], the violence score for each article was measured using NPMI as the measure of similarity which was proved to be the best. However, for measuring relief, CBOW showed the best results.

Improving Factor Scores

Three steps were performed to further improve the relief and violence scores of the documents:

1. **Expanding Seed Words**

   The primary set of seed words given by experts was expanded using a set of articles related to relief [‡], and the most frequent words shared among these articles were added to the set of seed words as well as the names of popular relief agencies and NGO organizations.

2. **Filtering Seed Words**

[‡]globalcorps.com/jobs/ngolist.pdf

One issue with the set of seed words was that some words related to relief were also related to other factors. Eg, 'death' was one of the words related to relief which was also found among the seed words related to violence. According to the fact that the set of seed words were generated manually and the words were picked based on their relevance to relief regardless of relevance to other factors, some seed words were not good representatives of relief. In this step, the seed words were filtered and only the words more related to relief than other factors were kept.

3. **Using Weighted Seed Words**

   Now we have the seed words that are most relevant to relief than to any other factor. But still some seed words are relatively more related to relief than other seed words. As an example, 'UN', 'relief', 'aid' or 'humanitarian' are more related than 'tent', 'disease' or 'supplies'. To consider this fact in our computations, we manually assigned a weight to each seed word. More relevant words to relief got higher weights. A subset of the final set of seed words after performing the three aforementioned steps and their weights are shown in Table 3.2.

Seed Words	Weight
Effort, relief, humanitarian, emergency, organization, agency, aid, donor, campaign, charity, assist, voluntary, motivation, rebuild, establish, UNICEF, NGO,UN	2
Community, epidemic, vaccination, nutrition, supplies, protect, shelter, cloth, food, money, water	1

Table 3.2: Seed words and their weights.

The similarity between a word (w) and a factor is then calculated using the

following updated formula:

$$\text{A set of } m \text{ seed words for factor } F : Y = y_1, y_2, ..., y_m$$

$$Similarity(W, F) = \frac{\sum_{k=1}^{m} Similarity(w, y_i) \times weight(y_i)}{\sum_{k=1}^{m} weight(y_i)} \tag{3.3}$$

In this research we are mainly focused on forced migration due to socio-political conflicts, thus, we used *Similarity with Seed-Words* technique to extract violence scores from news articles. We also used it to extract relief too, but, the other techniques introduced in this chapter are only focused on detecting violence.

3.2.2 Event Detection based Factor Extraction (ED-FE)

This technique intends to extract the events from news articles and then distinguish the violent events and use them to define a degree of violence for each date. An event is defined as an incident happening in a specific location involving entities, time and location. The task of event detection consists of event trigger extraction and event arguments extraction.

- **Event Trigger Extraction**: Event trigger is a word or phrase in the sentence, which evokes an event and carries the most information about it. For example, in the sentence: "Extremists clashed soldiers in Iraq on Friday", "clashed" is the event trigger. Event extraction task aims to detect whether or not a sentence includes an event trigger and what the trigger word's type is.

- **Event Arguments Extraction**: Event arguments extraction aims to extract the information about entities (locations, organizations and people), times and the role they play in an event. For example, in the aforementioned sentence, "extremists" and "soldiers" are the event arguments.

Training Data

The ACE2005 corpus[§] is a publicly accessible dataset which includes labels for the event triggers and arguments for a large set of sentences and is used for training purposes. ACE2005 event detection task, defines 8 event types and 33 sub-types[¶] for the whole corpus of sentences and assigns one event type to each event trigger for each sentence. Table 3.3 shows the predefined event types and sub-types.

Event Types	Event Sub-types
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End,Org
Conflict	Attack, demonstrate
Contact	Meet, Phone-write
Personnel	Start-position, End-position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 3.3: ACE2005 predefined event types and sub-types.

Furthermore, seven types of entities are identified in ACE2005: Person, Organization, Location, Facility, Weapon, Vehicle and Geo-Political Entity (GPEs). Each type is further divided into subtypes (for instance, Person subtypes include Individual, Group and Indefinite).

[§]https://catalog.ldc.upenn.edu/ldc2006t06

[¶]https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

```
<event ID="CNN_CF_20030303.1900.06-2-EV3" TYPE="Conflict" SUBTYPE="Demonstrate" MODALITY="Other"
  <event_argument REFID="CNN_CF_20030303.1900.06-2-E2" ROLE="Place"/>
  <event_argument REFID="CNN_CF_20030303.1900.06-2-E1" ROLE="Entity"/>
  <event_mention ID="CNN_CF_20030303.1900.06-2-EV3-1">
    <extent>
      <charseq START="725" END="767">200,000
people start protesting in Pakistan</charseq>
    </extent>
    <anchor>
      <charseq START="746" END="755">protesting</charseq>
    </anchor>
    <event_mention_argument REFID="CNN_CF_20030303.1900.06-2-E2-10" ROLE="Place">
      <extent>
        <charseq START="760" END="767">Pakistan</charseq>
      </extent>
    </event_mention_argument>
    <event_mention_argument REFID="CNN_CF_20030303.1900.06-2-E1-9" ROLE="Entity">
      <extent>
        <charseq START="725" END="738">200,000
people</charseq>
      </extent>
    </event_mention_argument>
  </event_mention>
</event>
```

Figure 3.4: An example of ACE2005 annotated corpus.

Figure 3.4 shows an example of ACE2005 annotated corpus for the sentence "200,000 people start protesting in Pakistan" *Protesting* is labeled as the trigger word with type *Conflict* and sub-type *Demonstrate*, and *Pakistan* and *200,000 people* are detected as its arguments (place and entity).

We examined event extraction methods introduced by [43, 21] and [23]. The method resulting in the best outcome according to the further evaluation explained at the end of this chapter was the one introduced by [23]. We used this method to train an event detection model on ACE2005 dataset and then apply it on our corpus of news articles to extract events. This approach jointly extracts events and entities within a document context. The learning problem is decomposed

into three tractable sub-problems: learning structures for a single event, learning event-event relations, and learning for entity extraction. Two objective functions are defined to solve the first two sub-problems and L-BFGS is used to optimize the training objectives. For entity extraction, a standard linear-chain Conditional Random Field (CRF) is trained. After all, these learned probabilistic models are integrated into a single model to jointly extract events and entities across a document.

Defining ED-FE Violence Scores

As this research is focused on detecting socio-political conflicts for which violence is the main reason, we particularly used this technique to detect violent events. After applying this method on our corpus of news articles to extract all event triggers and event arguments, the event triggers of types *injure, die* and *attack* were selected. We call these events the *violent events*. After that, four procedures were proposed to calculate the degree of violence.

1. **ED-FE-violent**: This violence score is computed by dividing the number of violent events for each date by the total number of events for that date.

2. **ED-FE-attack**: This violence score is computed by dividing the number of events with type *Attack* for each date by the total number of events for that date.

3. **ED-FE-Die**: This violence score is computed by dividing the number of events with type *Die* for each date by the total number of events for that

date.

4. **ED-FE-Injure**: This violence score is computed by dividing the number of events with type *Injure* for each date by the total number of events for that date.

As we want to model the population movements, we need to measure the impact of violence on people, which causes the movement. The degree of violence somewhat depends on the scale of other events happening at the same time. Thus, we put the total number of events as the denominator in the formula, to get a scale of the impact of violence on people.

3.2.3 Topic Detection based Factor Extraction (TD-FE)

This technique was introduced by [44] and uses topic modeling methods to analyze a large corpus of news articles to extract coherent topics which are further used to compute violence scores for specific dates. Topic modeling allows us to discover a distinct set of topics among the corpus by connecting the words with similar meanings to form topics. One benefit of using topic modeling for our research is that the various reports of one single incident gather together as one single topic. Thus, we can overcome the unbalanced coverage of news agencies over locations and incidents.

A topic is defined as a distribution over words while documents are distributions over topics. One important dimension of news articles is time. Articles reporting similar incidents, with the same publication date, are most likely to report the same incident. Also, news articles usually continue reporting about one

incident during the consecutive days. Thus, articles with same or close publication dates are most likely to form a single coherent topic about one particular subject.

To get high quality topics from our corpus of news articles, we explored a large corpus of news articles to extract the best topics representing factors of forced migration. In order to to gather all reports of one single incident into an individual topic, first we need to go though a step of classifying the news articles into time-windows. In this step, the news articles are analyzed and processed according to their time-windows. The length of the time-windows could vary from days to years. In this research, we chose the monthly time-windows for the topic analysis.

The main reason to use time-windows for topic analysis is that the topics with short time-period of existence, may be obscured by the generalized topics observed in the entire collection. Time-window based topic analysis also enables identifying granular and short-term topics, as well as generalized and long-term topics.

After dividing the original corpus into time-windows, we generate topic models for each time-window. We primarily generate topic models using LDA ([26]) and Non-negative Matrix Factorization (NMF) ([45]). Matrix factorization is a widely used approach for the analysis of high-dimensional data. The objective of NMF is to extract meaningful features from a set of non-negative sparse vectors. The NMF is successfully applied to different applications, such as image processing, hyper-spectral imaging, and text mining. Here we focus on the property of NMF to identify topics in a given set of documents and classify the documents among the underlying topics.

To identify the optimal number of topics k, we generate topics with the number of topics in the range of $k \in \{10, 50\}$ with an increment of two. To find the optimal k, we evaluate the quality of generated topics. One way to do so is to measure topic's semantic coherence. To calculate the topic coherence score, we use TC-W2V ([46]) and Unify framework ([47]). **TC-W2V** is a distributional semantics measure introduced by [46]. TC-W2V measure is based on the popular word2vec ([48]) word embedding technique. In this method, the coherence score is the mean pairwise Cosine similarity of the term vectors generated by Skip-gram model (Eq. 3.4).

$$TC - W2V = \frac{1}{\binom{N}{2}} \sum_{j=2}^{N} \sum_{i=1}^{j-1} similarity(wv_j, wv_i)) \tag{3.4}$$

Another recent work on topic coherence measure is the unifying framework proposed by [47]. The framework represents the coherence measures as a composition of parts, where the objective is to achieve higher correlation with human judgments. The framework has four segments at its core: segmentation of words subsets, probability estimation, confirmation measure, and aggregation.

We finally select the topic model resulting in the highest topic coherence generated by NMF model based on the unify score. At this step, we have generated and evaluated the coherence of the topic models for each time-window. For each of the topics, we generate a *topic-document* to represent the topic.

Topic-documents are generated by ranking the top keywords occurring in the topic according to their probability of appearing in the topic. The number of topics per time window may vary since we are only using the topics with best

coherence scores in the range of k.

Defining TD-FE Violence Scores

The generated *topic-documents* form a new corpus with reduced dimensionality. Our next step is to label a topic-document with one of these categories: violence/ terrorism, relief, economic issues, political conflicts, refugee crisis and environmental issues. This step can be done automatically using a topic labeling method. But we did it manually with the help of social scientists to ensure the quality of the topic labels. That is because the quality of the labeled data directly affects the quality of the forced displacement prediction model. The output of this stage is a set of labeled monthly topic-documents.

The violence score for each month is then defined by the total number of *violence/terrorism* topics for each month divided by the total number of topics for that month.

The division in the formula is justified using the same intuition from previous section regarding that the scale of violence is relevant to the scale of other events or incidents happening at the same time.

3.3 Emotion Detection

In order to study the effect of events and incidents on people, we refer to news articles, a good source for this sort of information. However, there is no computational method to measure the impact of incidents or events on people in terms of measuring how people react to these incidents and how they feel about everything

that is happening around them. People surrounded by disappointing or fearful news are more potential candidates for migration. The most useful techniques for such task are emotion detection methods which assign an emotion from a predefined set of emotion categories to news articles.

We used the SECO (Selective Co-occurrences) method introduced by [49] to extract six emotion categories from our corpus of news articles, including *anger, disgust, fear, happiness, sadness* and *surprise*. SECO is a sentence-level emotion detection method which first analyses sentences in order to extract words that could have some emotional content. These words are called cue words and usually include nouns, verbs, adverbs and adjectives. Each emotion category (in our case 6 categories) is represented by a set of seed words. For example, seed words representing *happiness* may include {*happy, joy*} or seed words for *anger* may be {*angry, mad*}. SECO is based on the assumption that a cue word co-occurs with only one emotion's seed word within any particular window of text.

In situations where multiple seed words from multiple emotion categories are contained in a context window, a procedure should be followed to select the most appropriate seed word co-occurring with the cue word. This is because following the aforementioned assumption, only one seed word is co-occurring with each cue word.

Results indicated that the best procedure includes selecting the closest preceding seed word to the cue word. The association or relatedness of cue words to seed words is measured using NPMI. After that, for each word w, its emotion vector for a specific emotion is computed by measuring its association with all

the seed words. Then, the emotion vector of a sentence is computed by averaging the emotion vectors of all its cue words. After all, a sentence is labeled with the emotion category for which it gained the maximum emotion vector.

After we label news articles with one of the six emotion categories, we compute the amount of a specific emotion for a specific date (E_{ed}) using the following formula:

$$E_{ed} = \frac{\sum D_{de}}{\sum D_d} \tag{3.5}$$

D_d: Articles for date d.

D_{de}: Articles with emotion e and for date d.

E_{ed}: The degree of emotion e for date d where $e \in \{happiness, sadness, anger, surprise, fear, disgust\}$.

The final output of this stage of the framework is the trend of six emotions over time for a specific location of interest.

3.4 Building Prediction Models

We build three types of prediction models for forced migration: The first one is purely based on time-series analysis. The second one only uses violence and emotion scores to make predictions. And the third one uses violence and emotion scores added as input variables into the time series models.

1. *Pure time-series approach:* We build auto-regression models on the time-series of forced displaced populations. Auto-regression is a regression model

with lagged variables as input features and it is built based on the assumption that the previous values (i.e., lagged values) of a variable might affect the future values. An auto-regression model is built in this approach using the numbers of refugees in the previous time points as input features and it makes predictions without other external source.

2. *Pure factor-based approach:* Extracted violence and emotion scores are used as input features of regression models to predict the future number of refugees. This approach investigates whether or not the violence or emotion scores are adequate enough to make accurate predictions.

3. *Time-series with factor scores:* Violence and emotion scores and lagged variables are both used as inputs for the predictive regression models. Comparing this approach with the pure time-series approach allows us to investigate the effectiveness of detected violence and emotion scores in improving the accuracy of forced migration prediction.

After all, the best feature set for predicting future refugee movements is reported.

The regression model in the aforementioned steps can be built using traditional machine learning models like SVR, Ordinary Linear Regression, Random Forest Regression or advanced neural network models like LSTM, GRU or multi-layer Perceptron regression. The details about the structure and functionality of these methods is provided below.

3.5 Regression Models

3.5.1 Ordinary Linear Regression

All linear models are built based on the assumption that the target value (y) is a linear combination of the input variables (x). As shown in the following formula, if ŷ is the predicted value, then:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_m x_m \tag{3.6}$$

Vector $w = (w_1, ...w_m)$ is called the coefficients of the model. The most simple and basic linear model, to which we refer as ordinary linear regression in this research, fits a linear model with coefficients $w = (w_1, ...w_m)$ to minimize a loss function which is the *squared-error loss function* in this case. The formula for squared-error loss function is as follows:

$$\textit{Squared error loss function} = (y - \hat{y})^2 \tag{3.7}$$

So basically, the problem of fitting a linear regression model is in the following form:

$$\min_w {||\hat{y} - y||_2}^2 \tag{3.8}$$

While ŷ is calculated using equation (3.6).

Ordinary linear regression minimizes the residual sum of squares between the actual values in the dataset (y), and the predicted values by the linear approxi-

mation ($\hat{y}$).

3.5.2 Stochastic Gradient Descent

Stochastic gradient descent is an affective approach to fit linear models and find the optimum coefficients. SGD is an optimization algorithm and could be used to fit various linear regression models with different loss functions. In this research, we use the term *SGD* in our experiments to refer to a linear regression model with *Huber* loss function, optimized by stochastic gradient decent. Huber loss function is less sensitive to noise and outliers comparing to squared-error loss function and is defined as follows:

$$L_\delta(y, \hat{y} = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leqslant \delta, \\ \delta|y - \hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \tag{3.9}$$

3.5.3 Support Vector Regression

Support Vector Machines (SVM) are also used for regression problems while keeping all the characteristics of the algorithm such as the maximal margin. The Support Vector Regression (SVR) is an algorithm for regression problems, with the exact same principles as the SVM for classification. Like SVM, SVR is also based on the assumption that data points are linearly separable and attempts to map a linear function (hyperplane) to the input variables while the error is minimized and the margins are maximized. If data is not linearly separable, kernel functions are used to transfer the data points into a higher dimensional feature space in

which linear separation would be possible.

3.5.4 Random Forest for Regression

Random forests or random decision forests for regression problems are an ensemble learning method. They build multiple decision trees based on training set, and produce the outputs by computing the average predicted values of the individual trees ([50])([51]). Random decision forests use averaging to overcome the over-fitting problem of the trees on the training set. The loss function used for random forest regression model is *Square-error loss* as stated in formula 3.7.

3.5.5 Multi-Layer Perceptron

Multi Layer Perceptron (MLP) can be used for both classification and regression problems. MLP for regression problems is the same as that for classification problems, only that it has no activation function in the output layer, which can also be seen as using the identity function as activation function. Thus, the loss function is defined as *square error loss*, and the output is a set of continuous values.

We use an input window for each of the above methods. A window defines how many time-steps to consider as the input of the model. Window size changes from 1 to 5 in our experiments. If the window size is 1, it means that features belonging to time-step t (features $\in$ {lagged variable, violence scores}) are used as input values for the model to predict future time-steps. If the window size is 5, it means that features belonging to time-steps *t, t-1, t-2, t-3, t-4, t-5* are

used as input values. A bigger window size allows the method to consider more information from the past and intuitively make more accurate predictions. On the other hand, a bigger window size increases the dimension of the input vector and thus increases the number of the parameters of the model to be estimated.

3.5.6 Long Short Term Memory

Long Short Term Memory (LSTM), is another kind of neural networks used in our experiments. Before explaining LSTM and how it is used for regression, lets introduce recurrent neural networks (RNN) first, as LSTM is a type of RNNs. RNN is a type of neural networks suitable for processing sequential data. In typical neural networks, it is assumed that all the inputs and outputs are non-related and independent from each other, which is not the prefect assumption when dealing with sequential data. For example, in natural language modeling, when neural networks are used to predict the next word in the sentence, they will function better if they know the words that have appeared before. To solve this problem, the output of RNNs is dependent on the previous computations. RNNs are called '*recurrent*' because they apply the same function on all the elements of the sequential input. RNNs have a type of memory which is capable of capturing the information about the previous inputs and previous calculations. Theoretically, RNNs should be able to capture the information about random long sequences, but in practice they are only capable of looking back a few steps over time. As a solution to this problem, Long Short Time Memory units (LSTM) were introduced by ([52]).

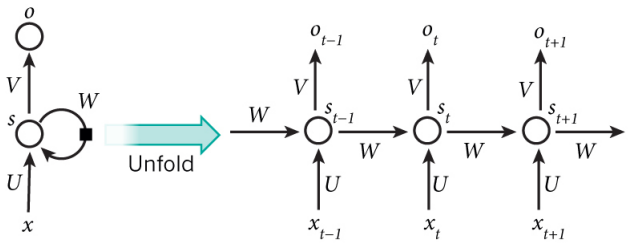

Figure 3.5: A recurrent neural network and the unfolding in time of the computation involved in its forward computation. ([53])

Figure 3.5 shows an RNN unfolded into a full network. Unfolding means writing down the computations and steps for the whole sequence of input over time. E.g., if the input is a sequence of 8 characters, the unfolded RNN will have the structure of a typical neural network with eight layers. The computations of a RNN are as follows:

- x_t is the input at time-step t. In language Modeling, for predicting the next word in a sentence, the words are given to the RNN one by one at each time-step. E.g., x_3 would be a vector representing the forth word of a sentence given to the network at time-step three.

- S_t is the memory or hidden state at time-step t. It is calculated based on the following formula:

$$s_t = f(Ux_t + Ws_{t-1}) \tag{3.10}$$

  s_t is calculated based on the current input and the previous hidden state. Function f is usually a nonlinear function like Rectified Linear Unit (ReLU) or Tangent Hyperbolic (Tanh).

- o_t is the output at time-step t. In the task of predicting the next word in a sentence, o_t would be the probability distribution over the vocabulary. $o_t = softmax(Vs_t)$

As stated above, the output at time-step t is only dependent on the hidden state at that time-step. We can think of s_t as sort of a memory which is capturing all the information from the previous steps. Unlike MLP whose knowledge from the past is limited to the size of the input window.

The other difference between RNNs and traditional deep neural networks is that RNN uses the same parameters (U, V, W) in all the steps while traditional neural networks use different parameters for each layer resulting in heavier computations.

LSTMs are the most popular type of neural networks that are more capable of capturing log-term dependencies than typical RNNs. LSTM has the same structure as RNN but differs in terms of calculating the hidden state. Plain RNNs have a very simple structured repeating module, however, the repeating module has a different and more complicated structure in LSTMs. This repeating module in LSTM is called memory unit and it is capable of controlling the information flowing through it. Memory unit keeps coded information from the past and by learning its parameters, the network learns how its memory should behave. In other words, the network can decide how much the historical information can effect the memory or how much impact the memory can have on the final output. This more control over the memory and the capability of longer-term dependencies in historical data, makes LSTMs a suitable choice for the purpose of this research

which is using previous knowledge of forced displacement to predict future migrations.

3.5.7 Gated Recurrent Unit

Gated Recurrent Units (GRU) are very much like LSTMs. They too, were created to solve the problem of better capturing long term dependencies. Like LSTM, GRU controls the flow of information too, but it does not use a memory unit. Instead, it exposes the full hidden content without any limitation.

GRU has less complex structure compared to LSTM, resulting in more efficiency in terms of computation.

All life is an experiment. The more experiments you make the better.

Ralph Waldo Emerson

4

Experimental Settings

4.1 Datasets

4.1.1 EOS dataset

The Expanded Open Source (EOS)* collection is a vast unstructured archive of over 700 million media articles gathered over the years by Georgetown University

*https://osvpr.georgetown.edu/eos

researchers. The collection is expanded by crawling over 20,000 Internet-based sources (e.g. news outlets, official agencies, blogs) and covers over 46 different languages. The news articles used in this research are filtered based on relativity to Iraq. The data set consists of 680,456 news articles spanning from January 2012 to May 2017.

4.1.2 UNHCR Statistical On-line Population Database

We used UNHCR[†] Refugee population statistics dataset to capture the number of refugees. This dataset contains data about forced displaced populations from 1999 to 2017 on a monthly basis. Information including the status of the population of concern (refugees, asylum seekers, internally displaced persons, etc.) and the origin and destination of the forced displaced persons is provided in the dataset. Figure 4.1 shows the total number of externally displaced Iraqis from 2012 to 2017.

For building prediction models we focus on a subset of UNHCR dataset related to Iraq from 2012 to 2017, as EOS dataset provides the news articles for this period of time. We aim to design models for predicting the future number of asylum-seekers for Iraq, thus, regression models are used in the experiments. Table 4.1 shows the statistical description of a subset of UNHCR dataset related to Iraqi refugees.

[†]United Nations High Commissioner for Refugees

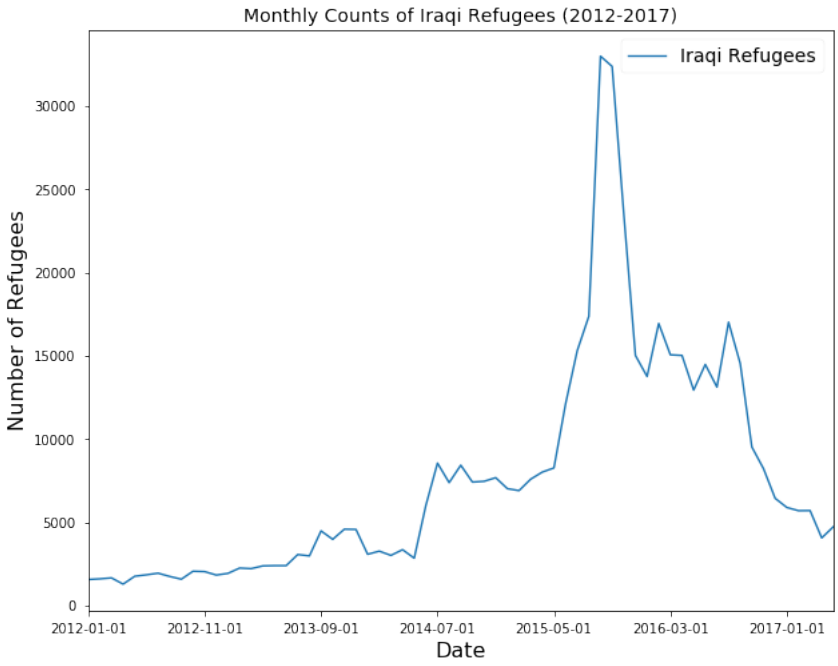

Figure 4.1: Total number of externally displaced Iraqis on a monthly basis from 2012 to 2017.

count	65
mean	7644.046154
std	6882.661314
min	1295
25%	2405
50%	5712
75%	9531
max	33006

Table 4.1: Statistical description of a subset of UNHCR dataset related to Iraq refugees (2012-2017).

4.2 Location Detection of News Articles

We applied our location detection method on EOS dataset to filter the articles related to Iraq. We used NLTK[‡], a toolkit for natural language processing in

[‡]http://www.nltk.org

Python, for Named Entity Recognition as part of our location detection algorithm. The total number of articles after location detection from 2012 to 2017 was approximately 500,000. For all the following experiments, we used the subset of EOS related to Iraq from 2012 to 2017.

4.3 Factor Extraction

4.3.1 Similarity with Seed Words (SWSW) violence scores

We applied SWSW on EOS news articles to extract violence and relief scores. For preliminary evaluation of the extracted scores, we followed the following steps:

Creating Evaluation Dataset

Following our previous work, [7], in which we created a manually labeled dataset for evaluating SWSW violence scores, we created an evaluation dataset to evaluate the extracted relief. To create the ground truth data set, three annotators manually labeled each news article with a score between 0 and 5, with 0 indicating the least and 5 indicating the most relevant to relief. We measured the inter-annotator agreement in terms of Cohen's kappa coefficient. Cohen suggested the Kappa result be interpreted as follows: values ≤ 0 as indicating no agreement and 0.01-0.20 as none to slight, 0.21-0.59 as fair, 0.60-0.79 as moderate, 0.80-0.90 as strong, and 0.90-1.00 as almost perfect agreement. The inter-annotator agreement between the three annotators stood at kappa=0.94.

Word Embeddings

300 dimensional word embeddings were trained on EOS dataset using Word2Vec[§] with negative sampling. Both CBOW and Skip-Gram methods were applied using this tool.

Validation of Relief Scores

We implemented a simple keyword matching system as the baseline method, and we compared and evaluated the three similarity measures using the manually annotated corpus. The baseline system simply gives similarity score=1 to any word that appears in the article and in the set of seed words, and score=0 if otherwise. Table 4.2 shows the person correlation coefficient between the manual and automatically obtained relief scores, which gives a value between [-1, 1] where 1 is total positive correlation, 0 is no correlation and -1 is total negative correlation. Enhancement in accuracy after three steps of improvement mentioned in (3.2.1) is also observable. Violence scores are directly reported from our previous work, [7].

Method	Pearson's Correlation			
	Relief		Violence	
	Before Improvements	After Improvements	Before Improvements	After Improvements
Baseline Method	0.542	0.571	0.682	0.730
CBOW	0.611	**0.672**	0.777	**0.878**
Skip-Gram	0.593	0.665	0.794	0.861
NPMI	0.607	0.670	0.800	0.842

Table 4.2: Pearson correlation coefficient between manual and automatically obtained relief and violence scores.

[§]https://radimrehurek.com/gensim/models/word2vec.html

All three methods of semantic similarity have outperformed the baseline by a considerable margin, however, CBOW has outperformed the other two methods. According to results obtained in our previous work, violence scores showed higher correlation with ground truth data (Pearson's correlation =0.8) but, relief showed Pearson's correlation of 0.67 with ground truth data. This can be explained by the fact that in this research, most of the articles were labeled as 0 or 5 rather than 2, 3 and 4 which means most of the articles had no mention of relief at all (95% of the articles were labeled as 0) and the ones that did relate to relief, were mostly labeled as 5. On the other hand, violence scores were distributed between 0 and 5 and this is because it is more sensible for human mind that how violent an event is than how much relief it carries. Relief is either present in a document or not, however, the degree of violence varies from a rubbery incident to a massive massacre. Therefore, as compared to violence scores, relief scores gained less Pearson correlation with annotated corpus.

Figure 4.2 shows the trend of violence extracted from EOS dataset using SWSW, from January 2012 to April 2017.

4.3.2 Event Detection-Factor Extraction (ED-FE) violence scores

As one news article might inform about more than one event, it is important to focus on the main event which is the most important incident reported in the article. According to [37], we can take into account our knowledge about the structure of news articles and assume that the main event often occurs in the title and the first sentence of the first paragraph of the articles. The rest of

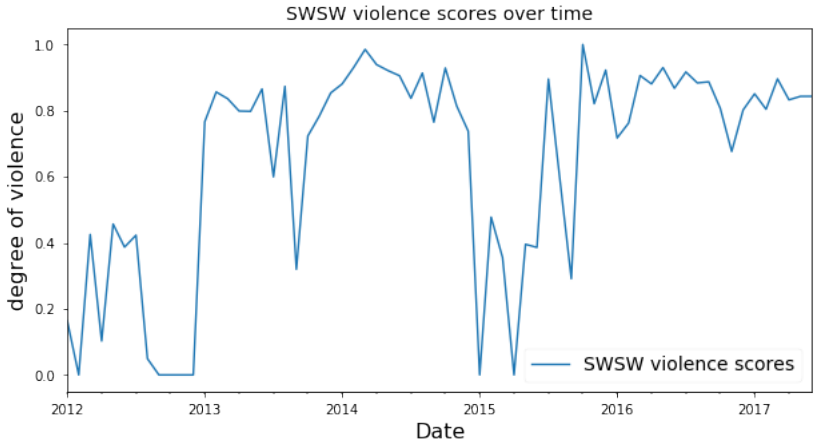

Figure 4.2: Extracted violence scores from EOS news articles using SWSW method.

the sentences and paragraphs commonly provide details about the main event or restate it in other words. Thus, we applied event detection algorithm on the first sentence of the first paragraph of news articles to extract event triggers and event arguments. We also removed the title of the articles as they often do not consist of full sentences, while the event detection algorithms work best on a context with the structure of a full sentence. Table 4.3 shows some example sentences from EOS dataset as well as the detected events. Detected events are in the form of an event trigger which has an event type, and event arguments. For violent events, event arguments commonly include the place or victim of the event.

Sentence	Events
Fighting continues in Iraq's Anbar Province as civilians flee.	Event type: Attack Event trigger: fighting Arguments: Place: Anbar_province
	Event type: Transport Event trigger: flee Arguments: Artifact: civilians
7 soldiers were killed and wounded by a roadside bomb south of Fallujah.	Event type: Die Event trigger: killed Arguments: Victim: soldiers, Instrument: bomb, Place: Fallujah
	Event type: Injure Event trigger: wounded Arguments: Victim: soldiers

Table 4.3: Some of the detected events from EOS dataset.

Figure 4.3 shows some event triggers and event arguments extracted from EOS dataset. For simple representation, the most frequent triggers or arguments were transfered into embedding space using Word2Vec, and then the dimensionality of the word vectors was reduced to 3 using Principle Component Analysis (PCA). The final vectors were plotted in a three dimensional space. Size of the circles indicates the frequency of the words. As observed, the most frequent event triggers include *killed, wounded, injured* and *clashes*, indicating that violent events are dominating the events related to Iraq. Also *Fallujah, Ramadi* and *Anbar* are observed among the most frequent words referring to the fact that conflicts were generally happening in these cities.

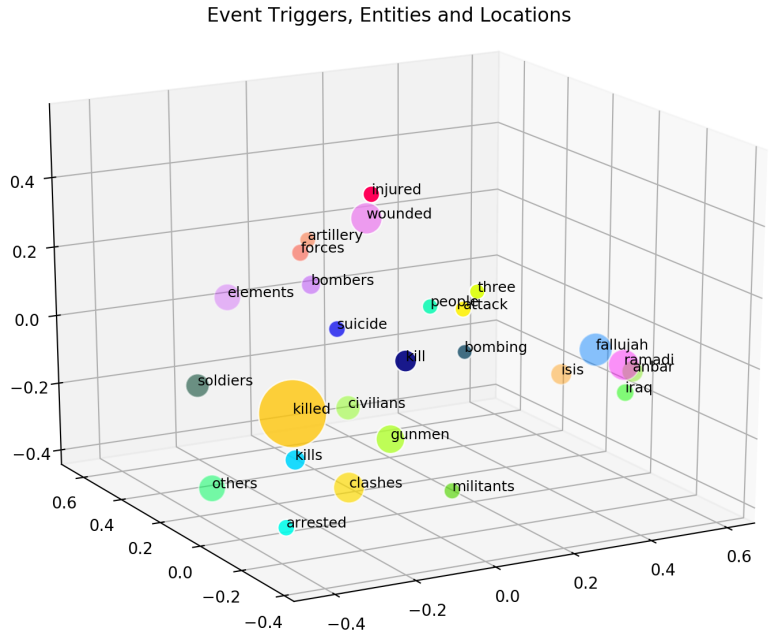

Figure 4.3: Some event triggers and arguments from a subset of EOS dataset.

Figure 4.4 is another representation of event triggers and event arguments.

The triggers or arguments associated with each other in form of one event, are connected with lines. Thickness of the lines indicates the frequency of the events including both words. As observed, *killed*, which is the most frequent event trigger, is strongly associated with *ISIS, Ramadi* and *Fallujah*, indicating the violent activities of extremists in those areas.

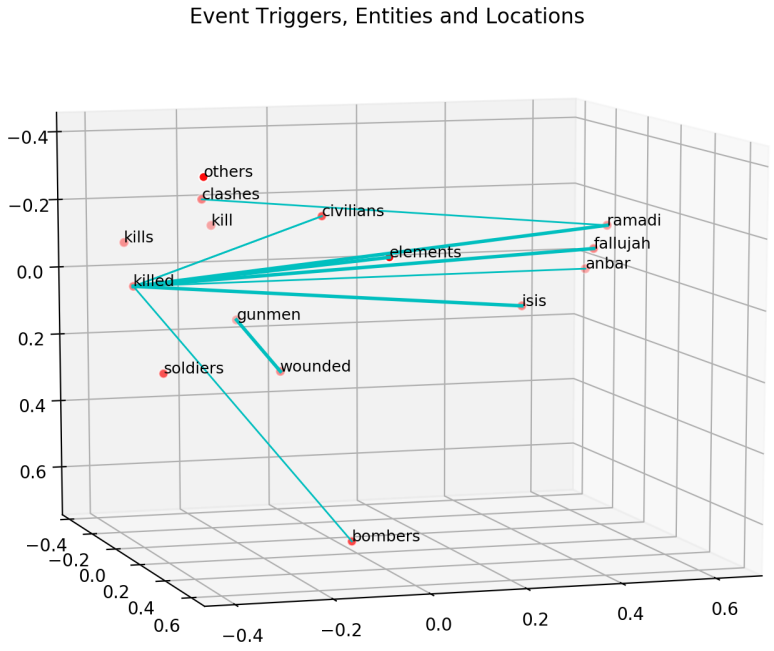

Figure 4.4: Some event triggers and arguments and the connections between them extracted from EOS dataset. Thickness of the lines indicates the number of times the words have co-occurred in one event.

After applying event detection algorithm on all the documents, articles with violent event types were selected and used to define ED-FE violence scores. Three violent event types were selected from ACE2005 event types, which were *attack, injure* and *die*. Four methods were proposed to measure violence using the ex-

tracted violent events (explained in 3.2.2).

Figure 4.5 shows the magnitude of ED-FE violence scores during 2012-2017, extracted from EOS dataset using ED-FE proposed method. The similar trend of ED-FE-Attack and ED-FE-Die is clearly visible. This might indicate that most of the attacks were companied by massive amounts of casualties. The quality of these violence scores are inspected and compared in the next chapter.

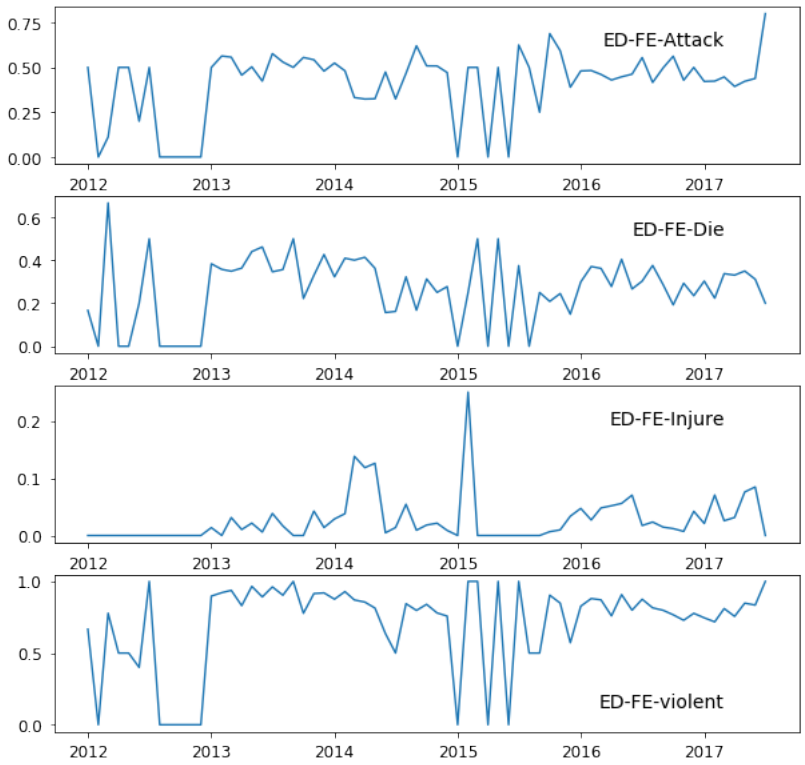

Figure 4.5: The trend of ED-FE violence scores during 64 months (January 2012-April 2017).

4.3.3 Topic Detection-Factor Extraction (TD-FE) violence scores

Defining Time-Windows

First step through analyzing the evolution of topics from EOS over time, is to separate the documents into time-window bins. To do so, we divided the EOS news articles into a set of sequential non-overlapping time-windows $\{T_1, ..., T_i\}$. Each time-window bin includes ordered documents, sorted by their publication date. We chose the length of time-windows to be one month, as UNHCR provides the information about refugees on a monthly basis.

Pre-processing

News articles were tokenized using Spacy[¶] library. Furthermore, bi-grams were extracted using the approach introduced in [54]. Depending on news articles, tri-grams might also be useful for the text analysis, however, we decided not to go further than bi-gram according to preliminary results. Bi-grams were the optimal level of text pre-processing in our case according to human judgment of the resulting topics.

Topic Modeling

After preprocessing news articles and separating them into time-window bins, topics were extracted for each month and 10 words best representing each topic were provided to two annotators who were asked to label topics in one of these

[¶]https://spacy.io/

categories: violence/terrorism, relief, economic issues, political conflicts, refugee crisis and environmental issues. The annotators labeled each topic separately, and then aggregate their results into the final labels to improve the quality of the labels.

We observed that the NMF topic modeling shows improvements in the topic modeling coherence scores. This result is in agreements with the results discovered in [55]. Another significant advantage of the NMF, compared to the probabilistic approaches, is the speed of finding topics. The matrix factorization tends to be faster than its counterpart probabilistic based approaches.

Table 4.4 shows some of the most coherent extracted topics from EOS dataset with their assigned category. In this table, topics are represented by their top ten words in terms of frequency.

Topic Label	Top 10 words (Sorted by their frequency in the topic)
Violence/Terrorism	killed, Baghdad, wound, car, attack, bomb, people, suicide, police, security,
Refugee Crisis	refugee, child, million, Jordan, UNHCR, Iraqi_refugee, humanitarian, people, flee, aid,
Economical Issues	oil, barrel, export, crude, company, market, energy, price, sanction, say
Relief	provide, support, food, UN, assistance, aid, information, facility, chemical, medical

Table 4.4: Examples of the extracted topics and their top ten words.

To calculate the TD-FE violence scores for a month, the number of violence-related topics in the month is divided by the total number of topics in that month. Figure 4.6 shows the extracted TD-FE violence scores from EOS news articles from January 2012 to April 2017.

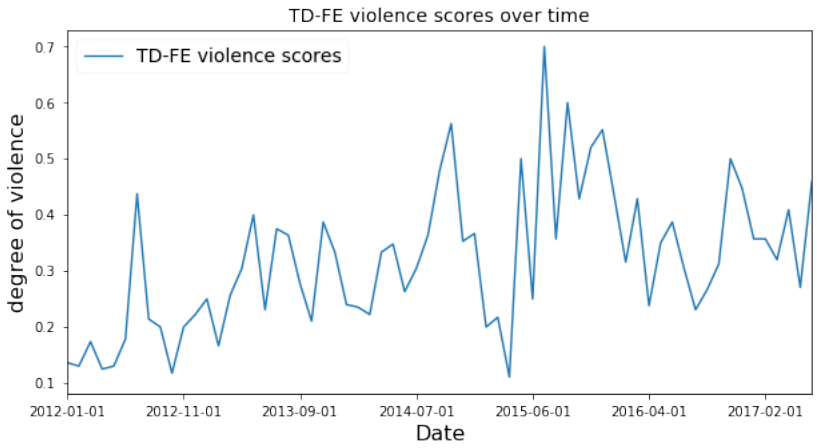

Figure 4.6: Extracted TD-FE violence scores from EOS dataset (January 2012 - April 2017).

4.4 Emotion Extraction

Six different emotions were extracted from EOS dataset. Figure 4.7 demonstrates the scores for all of the emotions during 64 months (January 2012 - April 2017). The contradiction between opposite emotions is clearly visible. Happiness decreases as fear, anger or disgust increase. Happiness is at its highest during the first 8 months of 2012, and after that it drastically reaches its lowest values. Negative emotions, such as fear, disgust and anger gradually increase as time goes on, indicating that people are dealing with more and more disappointing news as we get closer to the end of 2017. The effectiveness of these emotion scores in terms of improving prediction models for forced migration is reported in the next chapter.

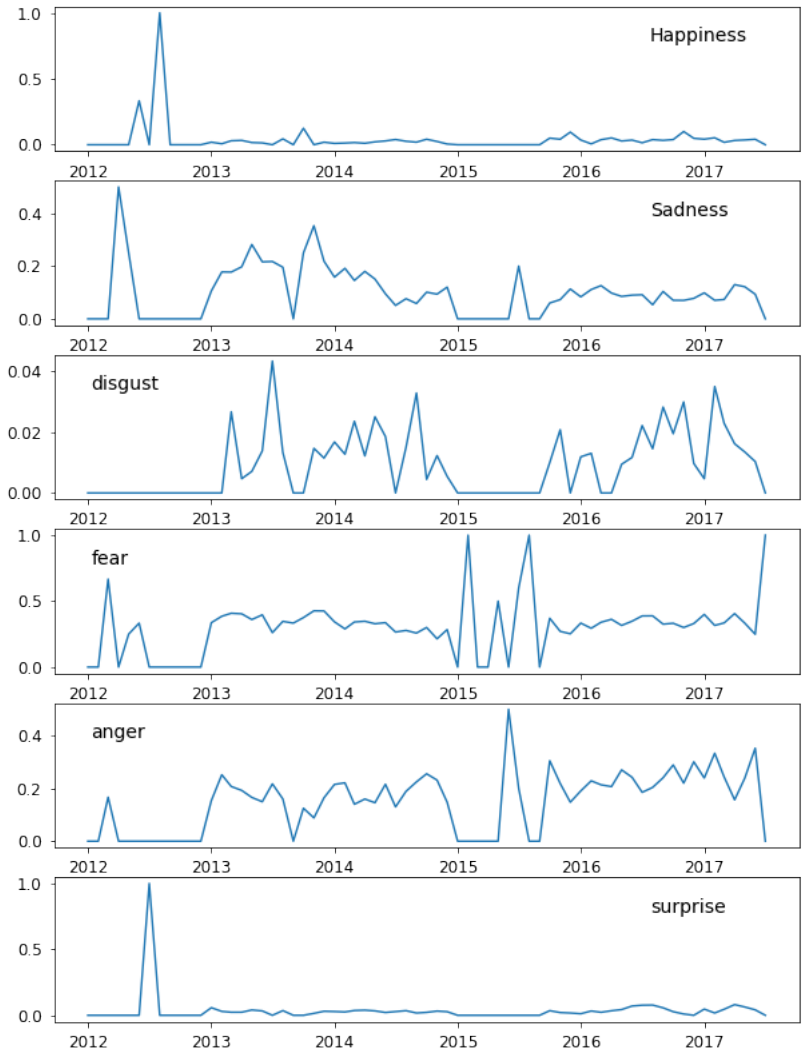

Figure 4.7: The trend of the extracted emotions from EOS dataset during 64 months (January 2012 - April 2017).

4.5 Prediction Models

4.5.1 Baseline

The baseline model for predicting forced migration is built using persistence algorithm (also called naive forecast), a very basic baseline method for supervised

machine learning regression problems. Persistence algorithm predicts the value for the future time-step (t+1) with the exact value seen at the previous time-step (t). Persistence algorithm can be implemented as a function that returns the same value given to it as the input. For example, the value mapped to January will be used as the input of the model and the output which is the predicted value for February, will be the exact value of January. Figure 4.8 shows the predicted values outputted by the baseline model for predicting t+1, from May 2016 to March 2017.

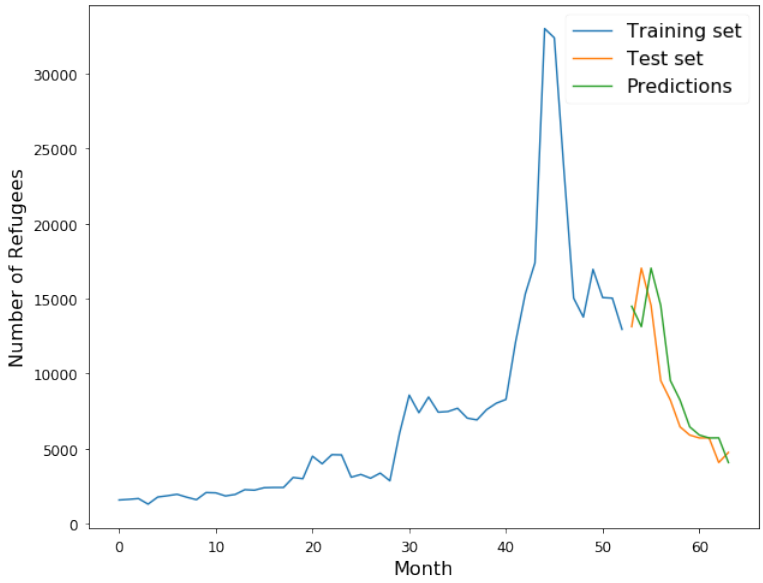

Figure 4.8: Persistence model's predictions for t+1.

4.5.2 Experimental Settings

We compare the three types of prediction models previously described in 3.4. In addition, we evaluate each prediction model in four *Settings*:

- Setting 1: Predicting value in time t+1
- Setting 2: Predicting value in time t+2
- Setting 3: Predicting value in time t+3
- Setting 4: Predicting value in time t+4

where the input to the model contains variables up to time t. This is because intuitively predicting t+2 is harder than t+1, and t+3 is harder than t+2 and so on. We would like to examine our prediction models in more difficult settings to be able to make sound conclusions about their prediction quality. Also, more time-steps a model can accurately predict in the future, more useful it will be for early warning systems in terms of preparing for refugee crisis. For example, the value of a model which can accurately (i.e., with an acceptable error rate) predict t+4, allows four months for governments to prepare for refugee crisis.

The UNHCR dataset is split into train set and test set. The training set includes 80% of the data (January 2012 - April 2016) and the test set includes the last 20% observations in UNHCR dataset (May 2016 - May 2017). Table 4.5 shows the test set for UNHCR dataset. The features used when predicting (t+n) time-step, all belong to time-steps before (t) and no information related to the time period of (t) until (t+n) is used, which makes predicting (t+n) harder as n increases. UNHCR for Iraq does not contain information for after April 2017, so we leave next time-steps untouched as there is no way to evaluate our predictions.

Historical Observations	Settings (Predicted time-steps)			
	t+1	t+2	t+3	t+4
Jan 2012 ... Apr 2016	May 2016	Jun 2016	Jul 2016	Aug 2016
Jan 2012 ... May 2016	Jun 2016	Jul 2016	Aug 2016	Sep 2016
Jan 2012 ... Jun 2016	Jul 2016	Aug 2016	Sep 2016	Oct 2016
Jan 2012 ... Jul 2016	Aug 2016	Sep 2016	Oct 2016	Nov 2016
Jan 2012 ... Aug 2016	Sep 2016	Oct 2016	Nov 2016	Dec 2016
Jan 2012 ... Sep 2016	Oct 2016	Nov 2016	Dec 2016	Jan 2016
Jan 2012 ... Oct 2016	Nov 2016	Dec 2016	Jan 2017	Feb 2016
Jan 2012 ... Nov 2016	Dec 2016	Jan 2017	Feb 2017	Mar2016
Jan 2012 ... Dec 2016	Jan 2017	Feb 2017	Mar 2017	Apr 2016
Jan 2012 ... Jan 2017	Feb 2017	Mar 2017	Apr 2017	May 2016

Table 4.5: The test set for UNHCR dataset. Number of refugees will be predicted in four settings (t+1, t+2, t+3 and t+4) based on the historical observations.

Root-Mean-Square Error (RMSE) is reported separately for each *Setting* and calculates the error of the predictions based on the test dataset. RMSE is calculated using the following formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (P_i - A_i)^2}{n}} \tag{4.1}$$

with

P = Predicted Value.

A = Actual Value.

n = Number of Samples.

A rolling forecast scenario, also called walk forward model evaluation is performed when evaluating the prediction results. Each time-step in the test set will be given to the model one at a time, the model predicts a value for the given time-step and then the actual value for that time-step will be accessible to the model to make the next predictions based on it. This is because we have more

than 10 instances in our test set and it is not possible to accurately predict all these corresponding values only based on the information available at the present time. E.g. in the *Setting t+4,* the rolling forecast scenario simulates a real world scenario when we make predictions for 4 months from now (t+4), and then the information about the next month (t+1) will be released (in our case news articles and IDP statistics) and then this information is used to make prediction for t+5.

4.6 Analyzing UNHCR dataset

We inspect UNHCR data for correlation between variables. Sometimes in time-series data, previous time-steps could be effective in predicting the future variables. This kind of correlation is called auto-correlation as it inspects the relationship between a variable and itself at a previous time-step. Figure 4.9 plots the observations at time-steps (t) versus the observations at the next time-steps (t+1). Obviously, there is some sort of correlation between the variables at subsequent time-steps.

In Sequential data, the observations at previous time-steps are called lagged variables. Figure 4.10 shows the auto-correlation calculated for different lagged variables using Pearson correlation. Lagged variable=n denotes the observed value at time-step (t-n). Pearson auto-correlation for lagged variable of 1 is 0.91. Solid and dashed lines in the picture represent the 95% and 99% confidence interval for the correlation values. The correlations of lagged variables less than 11 are statistically significant.

According to auto-correlation results, it makes sense to use auto-regression

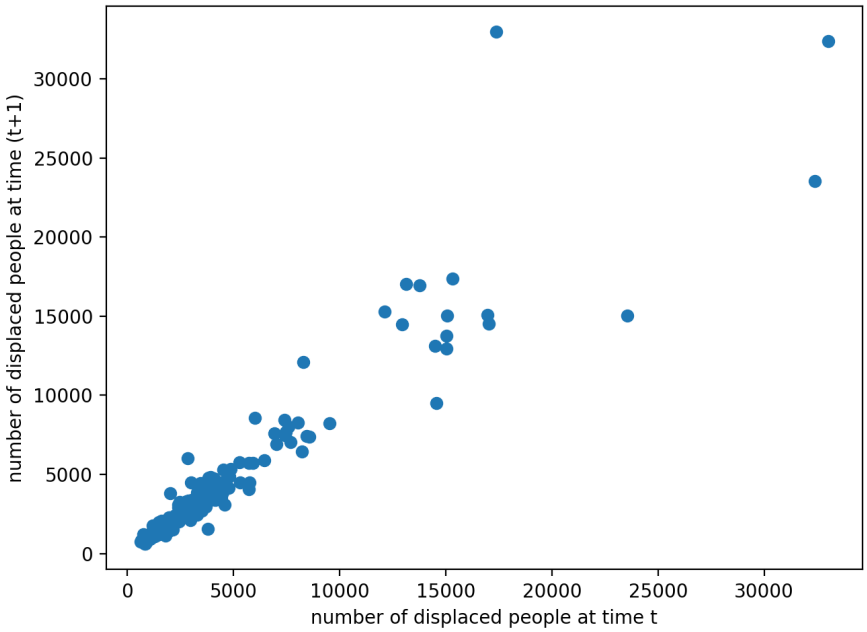

Figure 4.9: UNHCR dataset - Observations at time-step (t) versus the observations at (t+1).

models for predicting forced displacement. Auto-regression models are regression models that use lagged variables as inputs to predict the future time-steps. In the following experiments, each conducted using different regression models and features, lagged variables are also considered as potential input features. We have tested seven regression models previously explained in section 3.5 to predict future number of refugees. The effectiveness of the extracted features (violence and emotions scores) in improving these models is inspected. The results are represented in the following chapter.

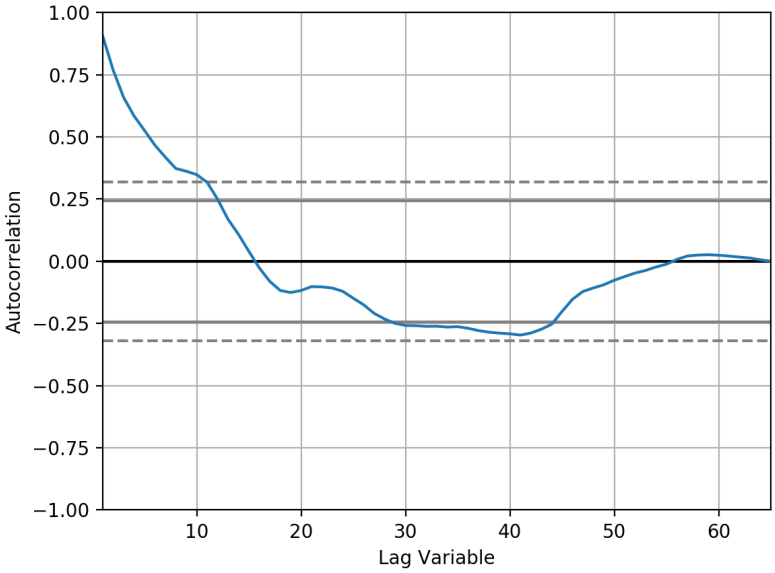

Figure 4.10: The auto-correlation plot for UNHCR dataset.

Knowing is not enough; we must apply. Willing is not enough; we must do.

Johann Wolfgang von Goethe

5

Results and Discussions

5.1 Evaluation of Prediction models

We evaluate our prediction models in four *settings* as described in (4.5.2): Predicting t+1, t+2, t+3 and t+4. Root Mean Square Error (RMSE) is reported separately for each setting and calculates the error of the predictions based on the test dataset. To evaluate the effectiveness and quality of the extracted features

(violence and emotions Scores) and to compare them against each other, we compare the predictive errors of the two types of regression models described in section (3.4), *pure time-series approach* and *time-series with factor scores* approach. If the error decreases when violence and emotion scores are added to time-series, it can be concluded that the added scores are effective elements in predicting forced migration. In the following sections, we have tested each feature (violence and emotion scores) individually with 7 different regression models, with the following configurations:

1. Random Forest Tree: Number of estimators= 450
2. Support Vector Regression (SVR): Kernel= Sigmoid
3. MLP Regression: Learning rate=0.00005, Optimizer= ADAM, Hidden layers=(5,5)
4. LSTM: Internal state size=30, Optimizer= ADAM
5. GRU: Internal state size=35, Optimizer= ADAM

We treated the window size as a parameter to be tuned individually for each model and all the following tables report the best outcome of each model.

5.1.1 Similarity with Seed Words (SWSW) Violence Scores

Table 5.1 shows the error rate of regression models on UNHCR dataset in terms of Root Mean-Square Error. Three types of regression models described in (3.4)

are built, using SWSW violence scores for *pure factor-based* and *time-series with factor scores* approaches.

The average error over the four *settings* is reported for each regression model. The decrease in average error with respect to *pure time-series approach* is marked in the table with down arrows. All the regression models showed improvement for *time-series with factor scores* over *pure time-series approach*, in terms of the average error (noted in the table with down arrows). The best performance was for SGD with RMSE of 2774, using both lag variables and SWSW violence scores as features of the model.

Furthermore, according to table 5.1, SWSW violence scores do not seem to be sufficient for making predictions (the performance of *pure factor-based* approach). However, *time-series with factor scores* approach is showing the best performance using the combination of lagged variables and SWSW violence scores as its input features.

5.1.2 Topic Detection-Factor Extraction (TD-FE) Violence Scores

Table 5.2 shows the error rate of regression models on UNHCR dataset in terms of Root Mean-Square Error. Three types of regression models described in (3.4) are built, using TD-FE violence scores for *pure factor-based* and *time-series with factor scores* approaches. The best performance was again achieved by *time-series with factor scores* approach using SGD, with RMSE of 3055. Also, like SWSW violence scores in previous section, TD-FE violence scores are not solely sufficient for making accurate predictions and they have to be combined with

Regression Models	Input Features		Predicted time-step (*Settings*)				
	lagged variable	SWSW violence scores	t+1	t+2	t+3	t+4	AVG(t+1,...t+4)
Ordinary Linear Regression	*		2094	2907	3522	3887	3102
		*	3956	3971	3993	3971	3972
	*	*	2154	2939	3523	2939	2888↓
MLP	*		2227	2827	3386	3649	3022
		*	4137	4103	4123	4103	4116
	*	*	2171	2966	3678	2966	2945↓
Random Forest	*		2794	5878	8353	10344	6842
		*	4711	4627	4835	4627	4700↓
	*	*	2398	6802	5072	6802	5268↓
SGD	*		2607	3172	3512	3611	3225
		*	5143	4975	4763	4975	4964
	*	*	**2094**	2877	**3248**	**2877**	**2774↓**
SVR	*		2510	2944	3263	3566	3070
		*	2094	3853	4229	3853	3507
	*	*	2094	2914	3373	2914	2823↓
LSTM	*		8524	6834	3478	3245	5520
		*	3999	2805	3320	3399	3380↓
	*	*	3416	4396	4687	3944	4110↓
GRU	*		2400	4272	6509	8986	5541
		*	4232	3758	4155	3900	4011↓
	*	*	2419	**2372**	5595	5499	3971↓

Table 5.1: Error rate of regression models in terms of RMSE. Input features set = { lagged variables, SWSW violence scores }.

lagged variables for better prediction performance.

5.1.3 Event Detection-Factor Extraction (ED-FE) Violence Scores

Table 5.3 shows the error rate of all the regression models in *pure time-series* setting. The best performance is for MLP-Regressor with RMSE of 3022. Other linear models (ordinary linear regression, SGD) are also ranked as top models in this case, indicating the high auto-correlation observed in UNHCR dataset (Previously reported in section (4.6)). The error increases when traversing from *Setting 1* (predicting t+1) to *Setting 4* (Predicting t+4), as expected according to the drop down of auto-correlation observed in (4.10).

Table 5.4 shows the RMSE of regression models in *time-series with factor scores* setting. The decrease in average error with respect to *pure time-series*

Regression Models	Input Features		Predicted time-step (*Settings*)				
	lagged variable	TD-FE violence scores	t+1	t+2	t+3	t+4	AVG(t+1,...t+4)
Ordinary Linear Regression	*		**2094**	2907	3522	3887	3102
		*	5330	4771	4693	4886	4920
	*	*	2135	2877	3373	3847	3058↓
MLP	*		2227	**2827**	3386	3649	3022
		*	4829	4563	4458	4201	4512
	*	*	2206	3364	3379	3905	3213
Random Forest	*		2794	5878	8353	10344	6842
		*	6349	6309	6090	5423	6042↓
	*	*	2353	5123	6533	6909	5229↓
SGD	*		2607	3172	3512	3611	3225
		*	5360	4983	4706	4166	4803
	*	*	2410	2935	3300	3575	**3055**↓
SVR	*		2510	2944	**3263**	3566	3070
		*	5165	4886	4688	4104	4710
	*	*	2510	3070	3388	3574	3135
LSTM	*		8524	6834	3478	**3245**	5520
		*	5056	5141	4571	4542	4827↓
	*	*	5772	2998	4352	6249	4842↓
GRU	*		2400	4272	6509	8986	5541
		*	5317	5035	4627	4422	4850↓
	*	*	2416	3363	5713	7472	4741↓

Table 5.2: Error rate of regression models in terms of RMSE. Input features set = { lagged variables, TD-FE violence scores }.

Regression Models	Predicted time-step (*Settings*)				
	t+1	t+2	t+3	t+4	Average (t+1,...t+4)
Ordinary Linear Regression	2094	2907	3522	3887	3102
MLP	2227	2827	3386	3649	**3022**
Random Forest	2794	5878	8353	10344	6042
SGD	2607	3172	3512	3611	3225
SVR	2510	2944	3263	3566	3070
LSTM	8524	6834	3478	3245	5520
GRU	2400	4272	6509	8986	5541

Table 5.3: The RMSE of regression models in *pure time-series* setting.

approach (as stated in table 5.3) is shown with down arrows, indicating the cases where adding ED-FE violence scores to input features has improved the performance. This improvement is observed for most of the models, indicating the effectiveness of ED-FE violence scores for predicting forced migration. ED-FE-violent is the only feature that adding it to the regression models has improved the performance for all of the 7 regression models.

Table 5.5 shows the average RMSE of all seven regression models using different violence scores achieved by SWSW, ED-FE and TD-FE techniques. On average, ED-FE-violent and TD-FE violence scores are both showing better performance for all t+1, t+2 and t+3, comparing to SWSW. To check whether the violence scores are able to improve the error of regression models significantly or not, we ran Wilcoxon signed-rank test separately for SWSW, ED-FE-violent and TD-FE violence scores. Wilcoxon signed-rank test tests the null hypothesis that the predictions using the *pure time-series approach* and the *time-series with factor scores* approach come from the same distribution. The test was run three times, between two approaches, with the second approach (*time-series with factor scores*) using a different set of violence scores (SWSW, ED-FE-violent and TD-FE violence scores) each time. The test's results indicate that ED-FE-violent and TD-FE violence scores have improved the prediction error significantly (rejecting the null hypothesis with p-value=0.0007 and p-value=0.03 respectively). However, SWSW violence scores were not able to significantly improve the prediction models' error (p-value=0.12).

5.1.4 Emotion Scores

Table 5.6 shows the RMSE of regression models in *time-series with factor scores* setting. The decrease in average error with respect to *pure time-series* approach (as stated in table 5.3) is shown with down arrows, indicating the cases where adding emotions to input features has decreased the error rate.

According to these results, *Disgust* and *Anger* are the most effective emo-

tions. All of the regression models using *Disgust* as an input feature, showed improvement over the case when *Disgust* was absent (demonstrated in table 5.3). Five out of seven models showed improvement when adding *Anger* to their input features. The best performance was for Multi-layer perceptron with RMSE of 2281, using both lagged variables and disgust scores as input features. We ran Wilcoxon signed-rank test separately for each emotion to check if emotion scores are capable of significantly improving the prediction models' performance. Wilcoxon signed-rank test tests the null hypothesis that the predictions using the *pure time-series approach* and the *time-series with factor scores* approach come from the same distribution. The test was run six times, between two approaches, with the *time-series with factor scores* approach using a different set of emotion scores each time. The test's results indicate that Disgust and Anger scores have improved the prediction error significantly (rejecting the null hypothesis with p-value=0.0009 and p-value=0.03 respectively). However, the other emotions were not able to significantly improve the prediction models' error.

5.1.5 The Final Model for Forced Displacement

For building the final model, a set of the most effective features were selected according to the previous experiments. All different combinations of these features were tested to build a final model with the best performance. After all, the least RMSE was achieved by MLP-regressor with feature set = { Lagged variable, ED-FE-violent scores, Disgust scores, Anger scores }. The best model was selected according to the average error over all the *Settings* (Average (t+1,...,t+4)).

Table 5.7 shows the RMSE of the final model on the test set as well as that of the baseline model. Our model considerably outperforms the baseline. This improvement over the baseline is most visible for *Setting 4* (predicting t+4), where the RMSE of the baseline is almost twice the RMSE of our final model.

Table 5.8 shows the Mean absolute percentage error (MAPE) of the final model on the test set as well as that of the baseline model. MAPE expresses the error as a percentage and is calculated using the following formula:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} |\frac{A_t - F_t}{A_t}| \tag{5.1}$$

Where A_t is the actual value and F_t is the forecast value.

Figure 5.1 shows the performance of the final model on UNCHR dataset, for *Setting 1* (predicting t+1). The error rate on both training and test sets are plotted as well as the actual values of the UNHCR dataset.

5.2 Analysis and Discussions

Our final model was built using a feature set including *Lagged variables, ED-FE-violent, Disgust scores* and *Anger scores*. Results indicate that our model beats the baseline with a considerable margin. According to these results and what was presented in table 5.5, we conclude that among all the proposed methods for measuring violence, ED-FE with ED-FE-violent scores outperforms the others. We believe that this is because ED-FE depends on detecting events which are more focused and detailed comparing to topics that the TD-FE violence detection method is based on. Also, SWSW violence detection relies on a predefined

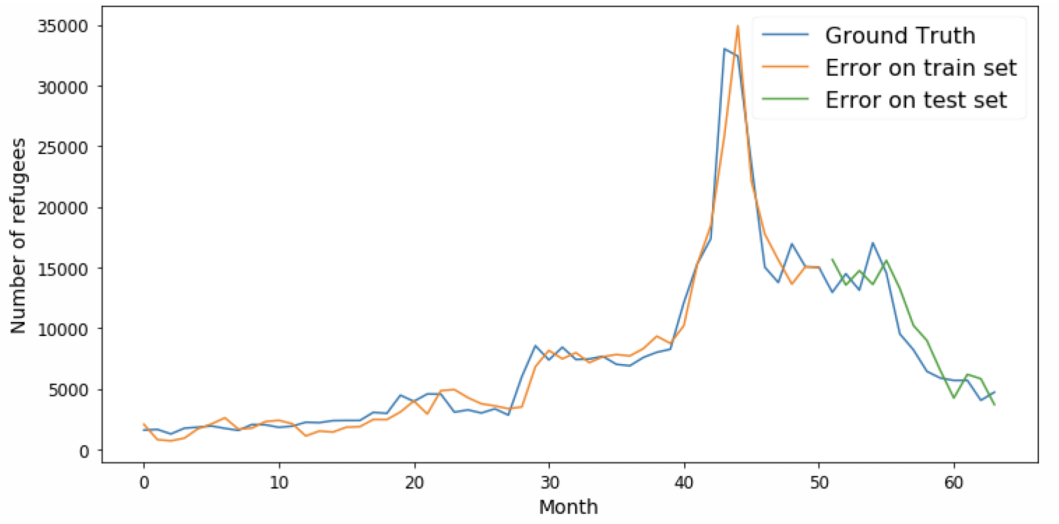

Figure 5.1: The predictions made by the final prediction model versus the actual values of UNHCR dataset.

set of seed words and the quality of SWSW violence scores directly depends on the quality of these seed words, while ED-FE does not have this disadvantage. Furthermore, the coverage of a violent incident by news agencies could be a good metric for measuring the degree of how violent that incident is. The bigger the size of the incident and its consequences, the more the number of news articles reporting it. ED-FE violence detection takes this information into account, while TD-FE completely ignores this material by gathering all the reports of a single incident into one unique topic.

It is important to note that going from *Setting 1* to *Setting 3*, the RMSE of the baseline increases gradually. This means that as we try to predict further time-steps in the future, the accuracy of the model decreases. On the other hand, the performance of our final model almost stays the same as we predict further time-steps in the future. This could be explained by the fact that when

predicting t+1, the lagged variable is probably the dominant feature, according to the very high auto-correlation of the UNHCR dataset for lagged variable=1 (91%). That is why our model and the baseline almost have similar performance for *Setting 1*. But as we intend to predict further time-steps in the future (e.g. t+4), the autocorrelation decreases and the model can no longer solely rely on lagged variables as the most effective features. Also, the model faces more missing information and longer gaps, which means that it needs to rely on other features to provide it with extra content to cover for the missing information (due to the gap between the present time-step (t), and the predicted time-step (t+4)). This is why the performance of our final model almost stays the same for all the *Settings*, while the error of the baseline increases gradually as we predict further time-steps in the future.

These results indicate that ED-FE violence scores are to some extent capable of providing the model with enough supporting information to make accurate predictions for future time-steps when we are facing gaps and missing information. Our intuitive justification for this incident is that our extracted violence scores are representing some sort of triggers for forced migration, meaning that they provide us with early signals of refugee movements. In other words, there is some gap in time between when these triggers take place and when refugee movements actually happen.

Furthermore, it is important to mention that relief scores were also tested in these experiments, but because of their poor performance we decided not to put the related results in this section. We believe that the challenges related to

extracting relief scores mentioned in section 4.3.1, are the main reasons that relief scores did not show as much improvement as violence and emotion scores. Also, its good to consider that generally other countries send relief and other sorts of emergency help after a disaster or harmful incident happens. Thus theoretically, relief occurs after the happening of the triggers of forced migration. So, extracted relief scores might not be very helpful in predicting future refugee movements.

Moreover, the average error for our final model is 2264 and it is within the tolerance range, considering that UNHCR time series has a mean of 7644 and standard deviation of 6882 (table 4.1). Results show that the violence or emotion scores are not sufficient to make predictions, however, combining them with lagged variables makes a powerful feature set for predicting forced migration. The impact of previous movements (lagged variables) on future movements might be due to the effect of the surrounding environment on people.

After all, the final model was created by MLP-regressor with a window of size 4. The reason that MLP-regressor outperformed the linear regression models might be due to its ability to learn complex non-linear functions. Furthermore, it is less complicated than GRU and LSTM and has fewer parameters, so it is easier to be trained especially on our relatively small dataset.

Models	Input Feature	Predicted time-step (*Settings*)				
		t+1	t+2	t+3	t+4	AVG (t+1,...t+4)
OLR *	ED-FE-attack	2128	2780	3404	3797	3027↓
	ED-FE-die	2117	2837	3507	3954	3101↓
	ED-FE-injure	**2107**	2562	3500	3875	3011↓
	ED-FE-violent	2121	2823	3479	3885	3077↓
MLP	ED-FE-attack	2180	2809	3154	**3418**	**2890**↓
	ED-FE-die	2299	2958	3504	3659	3105
	ED-FE-injure	2153	2872	3430	3685	3035
	ED-FE-violent	2136	2915	3305	3640	2999↓
RF †	ED-FE-attack	2311	5315	5785	6300	4927↓
	ED-FE-die	2447	5457	5676	5880	4865↓
	ED-FE-injure	2209	3516	5141	4416	3820↓
	ED-FE-violent	2382	3801	5793	4684	4165↓
SGD	ED-FE-attack	2475	2945	3150	3588	3039↓
	ED-FE-die	2574	2885	3066	3515	3010↓
	ED-FE-injure	2300	2783	**3003**	3423	**2877**↓
	ED-FE-violent	2216	2818	3185	3547	2941↓
SVR	ED-FE-attack	2356	2903	3290	3494	3010↓
	ED-FE-die	2335	2891	3267	3560	3013↓
	ED-FE-injure	2445	2906	3209	3459	3004↓
	ED-FE-violent	2352	2911	3277	3616	3039↓
LSTM	ED-FE-attack	3254	8316	4292	8356	6054
	ED-FE-die	5377	5386	7660	5638	6015
	ED-FE-injure	7164	10653	9412	11120	9587
	ED-FE-violent	3202	4158	5272	4151	4195 ↓
GRU	ED-FE-attack	2300	3123	2990	5275	3422↓
	ED-FE-die	2424	2898	4936	9389	4911↓
	ED-FE-injure	3210	4027	4394	6103	4433 ↓
	ED-FE-violent	2255	**2325**	3956	6677	3803 ↓

Table 5.4: The RMSE of the regression models in *time-series with factor scores* setting. Input features set = { lagged variables, ED-FE violence scores }.

Feature set	Predicted time-step		
	t+1	t+2	t+3
SWSW violence scores, lagged variables	2392	3609	4168
ED-FE-violent violence scores, lagged variables	2359	3021	3897
TD-FE violence scores, lagged variables	2828	3390	4291

Table 5.5: The average RMSE of all seven regression models using different violence scores achieved by SWSW, ED-FE and TD-FE techniques.

Models	Input Features	Predicted Time-step				
		t+1	t+2	t+3	t+4	AVG(t+1,...t+4)
OLR	fear	2253	2946	3598	3981	3194
	disgust	2064	2499	3026	3396	2746↓
	surprise	2093	2688	3264	3555	2900↓
	happiness	2090	2679	3257	3539	2891↓
	anger	2025	2832	3631	4361	3212
	sadness	2098	2841	3545	3937	3105
MLP	fear	2168	3079	3609	3938	3198
	disgust	2093	**2431**	**2600**	**2800**	**2481**↓
	surprise	2221	2940	3469	3787	3104
	happiness	2163	2946	3479	3790	3094
	anger	2025	3274	4170	4322	3447
	sadness	2192	2881	3457	3800	3082
RF	fear	2295	3344	4850	5239	3932↓
	disgust	2451	4451	5477	5236	4403↓
	surprise	2187	2870	4792	5066	3728↓
	happiness	2371	2811	5018	4474	3668↓
	anger	2071	5328	4826	5336	4390↓
	sadness	2206	3863	4869	5359	4074↓
SGD	fear	2232	2971	3271	3591	3016↓
	disgust	2078	2459	3120	3817	2868↓
	surprise	2295	2829	3183	5066	3343
	happiness	2279	2810	3184	3529	2950↓
	anger	2278	2963	3340	3605	3046↓
	sadness	2225	2874	3238	3580	2979↓
SVR	fear	2467	2965	3293	3592	3079
	disgust	2057	2723	3066	3437	2820↓
	surprise	2392	2917	3273	3563	3036↓
	happiness	2409	2919	3265	3571	3041↓
	anger	2249	2971	3352	3638	3052↓
	sadness	2294	2918	3303	3562	3019↓
LSTM	fear	2528	6126	3099	10393	5536
	disgust	3774	4990	7170	4807	5185↓
	surprise	4312	6783	6465	5573	5783
	happiness	4797	8086	5412	4824	5779
	anger	5327	2360	7556	4110	4838↓
	sadness	5146	4425	6000	4842	5103↓
GRU	fear	2953	3586	2662	5210	3602↓
	disgust	2347	2602	6153	5751	4213↓
	surprise	2281	3564	2820	2941	2901↓
	happiness	3005	5218	3935	6912	4767
	anger	**1861**	2773	4391	5295	3580↓
	sadness	2578	3803	3071	5970	3855↓

Table 5.6: The RMSE of the regression models in *time-series with factor scores* setting. Input features set = { lagged variables, ED-FE violence scores }.

Regression Model	Predicted time-step (*Settings*)				
	t+1	t+2	t+3	t+4	Average (t+1,...t+2)
MLP regression	1985	2313	2319	2442	**2264**
Baseline	2271	3517	4476	5736	4000

Table 5.7: The RMSE of the final model and the baseline.

Regression Model	Predicted time-step (*Settings*)				
	t+1	t+2	t+3	t+4	Average (t+1,...t+2)
MLP regression	13.6%	14.1%	14.6%	14.3%	**14.225**
Baseline	20.7%	34.4%	46.7%	59.4%	40.22%

Table 5.8: The MAPE of the final model and the baseline.

6

Conclusion

In this work, we presented a novel application of machine learning and natural language processing techniques to predicting forced migration based on news articles. We proposed a novel framework for processing and analyzing news articles to extract the factors of forced displacement and use them to build prediction models for forecasting future numbers of forced displaced people. In this frame-

work, we proposed two novel techniques called ED-FE and TD-FE for processing and analyzing news articles to extract the violence scores based on event detection and topic detection, respectively. We made comparisons between ED-FE, TD-FE and a state-of-the-art method called SWSW. Experiments demonstrate that both ED-FE and TD-FE outperform SWSW. Moreover, ED-FE was identified as the most effective technique for extracting violence scores among the three methods.

Also, we detected six human emotions from news documents, while *Anger* and *Disgust* proved to be the most useful ones for building prediction models. Our final prediction model was built using Multi-Layer Perceptron Regression with a feature set including ED-FE violence scores, Anger scores and Disgust scores. This model outperformed the baseline with a considerable margin. Results show that adding Anger, Disgust and ED-FE violence scores to the input features of prediction models significantly improves the prediction accuracy indicating that we can rely on violence scores detected from news articles as a useful factor for predicting forced displacement.

Furthermore, the performance of our proposed framework depends on the quality of the corpus of news articles in terms of complete and accurate coverage of the events happening inside the environment of concern. Unfortunately, the EOS dataset has many missing articles during the first six months of 2015, resulting in a negative impact on the quality of our model. This is observable in the sudden drop down in extracted violence scores during 2015 (Figures 4.2, 4.5, 4.6). We believe that our framework will show more accurate results using a dataset with better coverage.

Also, there are many other factors impacting refugee migration, which might not be extractable from news articles using our proposed framework, such as European Union's policy on accepting more refugees from middle east during 2015 resulting in the significant increase in the number of refugees during this year.

After all, we showed in this research that news articles are powerful resources for studying forced displacement and by effectively processing them we can extract useful features to build prediction models capable of forecasting future refugee movements.

As part of the future work, this research can be extended to extracting other factors of forced migration such as relief, economic instability or environmental threats from news articles. The effectiveness of the extracted factors in building prediction models could also be inspected and the useful factors could be integrated into the prediction model to make more accurate forecasts. Furthermore, TD-FE violence extraction can be more automatized using automatic topic labeling. One way to do this is to use a set of seed words to represent a topic label (such as violence, relief, or economic issues), measure the similarity between the set of seed words and a set of top-ranked keywords in a topic from the topic modeling result, and label the topic with the most similar topic label.

References

[1] UN Refugee Agency. Report of the united nations high commissioner for refugees. Covering the period 1 July 2015-30 June 2016, 2016.

[2] David Turton. Conceptualising forced migration. 2003.

[3] Roberta Cohen and Francis M Deng. Mass displacement caused by conflicts and one-sided violence: national and international responses. *SIPRI Yearbook*, page 15, 2009.

[4] Stephen Castles. Towards a sociology of forced migration and social transformation. *sociology*, 37(1):13–34, 2003.

[5] Susanne Schmeidl and J Craig Jenkins. Issues in quantitative modelling in the early warning of refugee migration. *Refuge: Canada's Journal on Refugees*, 15(4), 1996.

[6] Susanne Schmeidl. Exploring the causes of forced migration: A pooled time-series analysis, 1971-1990. *Social Science Quarterly*, pages 284–308, 1997.

[7] Ameeta Agrawal, Raghavender Sahdev, Heidar Davoudi, Forouq Khonsari, Aijun An, and Susan McGrath. Detecting the magnitude of events from news articles. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, pages 177–184. IEEE, 2016.

[8] Sheng-Hao Hung, Chia-Hung Lin, and Jen-Shin Hong. Web mining for event-based commonsense knowledge using lexico-syntactic pattern matching and semantic role labeling. *Expert Systems with Applications*, 37(1): 341–347, 2010.

[9] Fang Li, Huanye Sheng, and Dongmo Zhang. Event pattern discovery from the stock market bulletin. In *Discovery Science*, pages 35–49. Springer, 2002.

[10] Akane Yakushiji, Yuka Tateisi, Yusuke Miyao, and Jun'ichi Tsujii. Event extraction from biomedical papers using a full parser. In *Pacific Symposium on Biocomputing*, volume 6, pages 408–419, 2001.

[11] Yoko Nishihara, Keita Sato, and Wataru Sunayama. Event extraction and visualization for obtaining personal experiences from blogs. In *Symposium on Human Interface*, pages 315–324. Springer, 2009.

[12] Heng Ji, Ralph Grishman, et al. Refining event extraction through cross-document inference. In *ACL*, pages 254–262, 2008.

[13] Prashant Gupta and Heng Ji. Predicting unknown time arguments based on cross-event propagation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 369–372. Association for Computational Linguistics, 2009.

[14] Siddharth Patwardhan and Ellen Riloff. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 151–160. Association for Computational Linguistics, 2009.

[15] Shasha Liao and Ralph Grishman. Acquiring topic features to improve event extraction: in pre-selected and balanced collections. In *RANLP*, pages 9–16, 2011.

[16] David McClosky, Mihai Surdeanu, and Christopher D Manning. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1626–1635. Association for Computational Linguistics, 2011.

[17] Ruihong Huang and Ellen Riloff. Modeling textual cohesion for event extraction. In *AAAI*, 2012.

[18] Sunandan Chakraborty, Ashwin Venkataraman, Srikanth Jagabathula, and Lakshminarayanan Subramanian. Predicting socio-economic indicators using news events. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1455–1464. ACM, 2016.

[19] Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1127–1136. Association for Computational Linguistics, 2011.

[20] Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In *ACL (1)*, pages 73–82, 2013.

[21] Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. In *ACL (2)*, pages 365–371, 2015.

[22] Thien Huu Nguyen and Ralph Grishman. Modeling skip-grams for event detection with convolutional neural networks. In *EMNLP*, pages 886–891, 2016.

[23] Bishan Yang and Tom Mitchell. Joint extraction of events and entities within a document context. *arXiv preprint arXiv:1609.03632*, 2016.

[24] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, Jun Zhao, et al. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL (1)*, pages 167–176, 2015.

[25] Yifang Wei, Lisa Singh, Brian Gallagher, and David Buttler. Overlapping target event and story line detection of online newspaper articles. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 222–232. IEEE, 2016.

[26] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[27] David M Blei and John D Lafferty. Correlated topic models. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pages 147–154. MIT Press, 2005.

[28] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584. ACM, 2006.

[29] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.

[30] Shoaib Jameel and Wai Lam. An unsupervised topic segmentation model incorporating word order. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 203–212. ACM, 2013.

[31] Ruchi Hirat and Namita Mittal. A survey on emotion detection techniques using text in blogposts. *International Bulletin of Mathematical Research*, 2 (1):180–187, 2015.

[32] Edward Chao-Chun Kao, Chun-Chieh Liu, Ting-Hao Yang, Chang-Tai Hsieh, and Von-Wun Soo. Towards text-based emotion detection a survey and possible improvements. In *Information Management and Engineering, 2009. ICIME'09. International Conference on*, pages 70–74. IEEE, 2009.

[33] Gabriel Recchia and Michael N Jones. More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. *Behavior research methods*, 41(3):647–656, 2009.

[34] David B Bracewell. Semi-automatic creation of an emotion dictionary using wordnet and its evaluation. In *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, pages 1385–1389. IEEE, 2008.

[35] Chung-Hsien Wu, Ze-Jing Chuang, and Yu-Chung Lin. Emotion recognition from text using semantic labels and separable mixture models. *ACM transactions on Asian language information processing (TALIP)*, 5(2):165–183, 2006.

[36] Hema Krishnan, M Sudheep Elayidom, and T Santhanakrishnan. Emotion detection of tweets using naïve bayes classifier. *Emotion*, 2017.

[37] Jakub Piskorski, Hristo Tanev, Martin Atkinson, Eric Van Der Goot, and Vanni Zavarella. Online news event extraction for global crisis surveillance. In *Transactions on computational collective intelligence V*, pages 182–212. Springer, 2011.

[38] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[39] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009.

[40] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

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

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

[43] Reza Ghaeini, Xiaoli Z Fern, Liang Huang, and Prasad Tadepalli. Event nugget detection with forward-backward recurrent neural networks. *arXiv preprint arXiv:1802.05672*, 2018.

[44] Sadra Abrishamkar, Forouq Khonsari, Aijun An, and Jimmy Huang. Predicting forced population displacement using news articles. *Manuscript Submitted for Publication*, 2018.

[45] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, oct 1999. ISSN 0028-0836.

[46] Derek O 'callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. An Analysis of the Coherence of Descriptors in Topic Modeling. *Expert Systems with Applications (ESWA)*, 2015.

[47] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15*, pages 399–408, New York, New York, USA, 2015. ACM Press. ISBN 9781450333177.

[48] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[49] Ameeta Agrawal and Aijun An. Selective co-occurrences for word-emotion association. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1579–1590, 2016.

[50] Tin Kam Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

[51] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[52] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[53] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[54] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. 2013.

[55] Derek Greene and James P Cross. Exploring the Political Agenda of the European Parliament Using a Dynamic Topic Modeling Approach. *Political Analysis*, 2016.