NOAA Exploration

David Haycraft

2/25/2021

Synopsis

Sever Weather Events happen with an alarming frequency. In the United States NOAA tracks the damage in terms of human damage and economic damage. The data was consolidated into groups using a combination of string similarity and semantic similarity to identify the top weather events related to property damage and population health. There are important regional factors that may be related to the occurrence but according to this study tornado events are the most severe in terms of fatalities and injuries by a wide margin. A more predictable and possibly easier to control event is deaths from heat and excessive heat which were the second and third most common events related to fatalities. Tornadoes are also the top cause of property damage closely followed by thunderstorms with accompanying winds. Finally, top cause of crop damage is hail by a very wide margin.

Data Processing

The data was loaded directly from the NOAA Storm data website and was then read in as csv.

```
temp <- tempfile()
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2",temp)
noaa_df <- read.csv(temp, header=TRUE)
file.remove(temp)
```

```
## [1] TRUE
```

Preprocessing for Events:

- Normalize strings by using make all letters upper case and trim white space
- Remove punctuation and replace with spaces
- Remove numbers
- Convert TSTM abbreviation to THUNDERSTORM

Each serves to cleanse the text before we attempt fuzzy matching and semantic similarity matching. When the above is complete in order to reduce the noise in the EVTYPE variable we will find the distance between all unique EVTYPEs for string distance and semantic similarity based on the Universal Sentence Encoder from Google. After calculating each of the distance matrices was normalized to have distances scaled from 0 to 1. Now since the two distances distances are on the same scale we will take average distance will then be used as an input to hierarchical clustering. Finally, after experimenting with different heights for cutting the dendogram produced by hierarchical clustering it will be used to consolidate groups that are like one another. Once the groups are established the most frequently appearing member of each group is designated as the label for the groups.

```
library(dplyr)
library(stringdist)
library(tidyr)
library(tfhub)
library(plotly)
library(ggplot2)
library(DT)
library(gridExtra)

# need to install tensorflow-hub into the conda environment

embeddings <- tfhub::hub_load("https://tfhub.dev/google/universal-sentence-encoder/4")
noaa_df <- noaa_df %>% mutate(EVTYPE = toupper(trimws(EVTYPE, which="both")),
                              EVTYPE=gsub("[0-9]", "", EVTYPE),
                              EVTYPE = gsub("[[:punct:]]", " ", EVTYPE),
                              EVTYPE = gsub("TSTM", "THUNDERSTORM", EVTYPE))

scale_zero_one <- function(x){
  max_val <- max(x)
  min_val <- min(x)
  (x-min_val)/(max_val-min_val)
}


unique_events <- unique(noaa_df$EVTYPE)
embed_unique <- as.data.frame(as.matrix(embeddings(as.array(unique_events))))
dist_embed <- dist(embed_unique)
dist_embed_scale <- scale_zero_one(dist_embed)
#find the length of the longer of two strings in each pair
pwmax <- combn(nchar(unique_events),2,max,simplify = T)
dist_events <- stringdistmatrix(unique_events)/pwmax
dist_events_scale <- scale_zero_one(dist_events)

dist_avg <- (dist_embed_scale+dist_events_scale)/2

event_clust <- hclust(dist_avg)
# plot(event_clust)
cut_labs <- cutree(event_clust, h=.6)
event_groups <- data.frame(cut_labs, unique_events) %>% arrange(cut_labs)

event_freq <- noaa_df %>% group_by(EVTYPE) %>%
                          summarise(cnt = n())


event_grp_cnts <- left_join(event_groups, event_freq, by=c("unique_events"="EVTYPE"))

max_event_by_group <- event_grp_cnts %>% group_by(cut_labs) %>%
                                         arrange(desc(cnt)) %>%
                                         filter(row_number()==1) %>%
                                         select(unique_events, cut_labs) %>%
                                         rename(grp_event = unique_events)

events_map <- left_join(event_grp_cnts, max_event_by_group, by=c("cut_labs"="cut_labs")) %>%
```

```
                 select(unique_events, grp_event)


noaa_df <- left_join(noaa_df, events_map, by=c("EVTYPE"="unique_events"))


noaa_df <- noaa_df %>% mutate(CROPDMGEXP = toupper(CROPDMGEXP),
                              PROPDMGEXP = toupper(PROPDMGEXP),
                              CROPDMGEXP = case_when(CROPDMGEXP=="B"~9,
                                                     CROPDMGEXP=="M"~6,
                                                     CROPDMGEXP=="K"~3,
                                                     TRUE~0),
                              PROPDMGEXP = case_when(PROPDMGEXP=="B"~9,
                                                     PROPDMGEXP=="M"~6,
                                                     PROPDMGEXP=="K"~3,
                                                     is.numeric(as.numeric(PROPDMGEXP))~as.numer
ic(PROPDMGEXP),
                                                     TRUE~0),
                              CROPDMG = CROPDMG*10^CROPDMGEXP,
                              PROPDMG = PROPDMG*10^PROPDMGEXP)




human_damage <- noaa_df %>% group_by(grp_event) %>%
                            summarise(TOT_FATAL= sum(FATALITIES, na.rm=TRUE),
                                      TOT_INJURED = sum(INJURIES, na.rm=TRUE))


fatal_damage <- human_damage %>% filter(TOT_FATAL!=0) %>%
                                 arrange(desc(TOT_FATAL)) %>%
                                 top_n(20)


fatal_damage$grp_event <- factor(fatal_damage$grp_event, levels=fatal_damage$grp_event)

fatal_plot <-  ggplot(fatal_damage, aes(grp_event, TOT_FATAL)) +
                 geom_bar(stat = 'identity')+
                 theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
                 labs(title="Top 20 Weather Events by Number of Fatalities",
                      x="Weather Event",
                      y="Total Fatalities")




injury_damage <- human_damage %>% filter(TOT_INJURED!=0) %>%
                                 arrange(desc(TOT_INJURED)) %>%
                                 top_n(20)


injury_damage$grp_event <- factor(injury_damage$grp_event, levels=injury_damage$grp_event)

injury_plot <-  ggplot(injury_damage, aes(grp_event, TOT_INJURED)) +
                 geom_bar(stat = 'identity')+
```

```
                 theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
                 labs(title="Top 20 Weather Events by Number Injured",
                      x="Weather Event",
                      y="Total Injured")


econ_damage <- noaa_df %>% group_by(grp_event) %>%
                            summarise(TOT_PROPDMG= sum(PROPDMG, na.rm=TRUE),
                                      TOT_CROPDMG = sum(CROPDMG, na.rm=TRUE))


prop_damage <- econ_damage %>% filter(TOT_PROPDMG!=0) %>%
                                 arrange(desc(TOT_PROPDMG)) %>%
                                 top_n(20)


prop_damage$grp_event <- factor(prop_damage$grp_event, levels=prop_damage$grp_event)

prop_damage_plot <-  ggplot(prop_damage, aes(grp_event, TOT_PROPDMG)) +
                 geom_bar(stat = 'identity')+
                 theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
                 labs(title="Top 20 Weather Events by Total Property Damage",
                      x="Weather Event",
                      y="Total Property Damage")




crop_damage <- econ_damage %>% filter(TOT_CROPDMG!=0) %>%
                                 arrange(desc(TOT_CROPDMG)) %>%
                                 top_n(20)


crop_damage$grp_event <- factor(crop_damage$grp_event, levels=crop_damage$grp_event)

crop_damage_plot <-  ggplot(crop_damage, aes(grp_event, TOT_CROPDMG)) +
                 geom_bar(stat = 'identity')+
                 theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
                 labs(title="Top 20 Weather Events by Total Crop Damage",
                      x="Weather Event",
                      y="Total Crop Damage")
```

```
datatable(events_map)
```

Show 10 **entries** **Search:**

	unique_events	grp_event
1	TORNADO	TORNADO
2	TORNADO F	TORNADO

	unique_events	grp_event
3	TORNADOS	TORNADO
4	TORNADO WATERSPOUT	TORNADO
5	TORNADOES	TORNADO
6	TORNADO DEBRIS	TORNADO
7	THUNDERSTORM WIND	THUNDERSTORM WIND
8	THUNDERSTORM WINDS	THUNDERSTORM WIND
9	THUNDERSTORMS WINDS	THUNDERSTORM WIND
10	THUNDERSTORM WINDS	THUNDERSTORM WIND

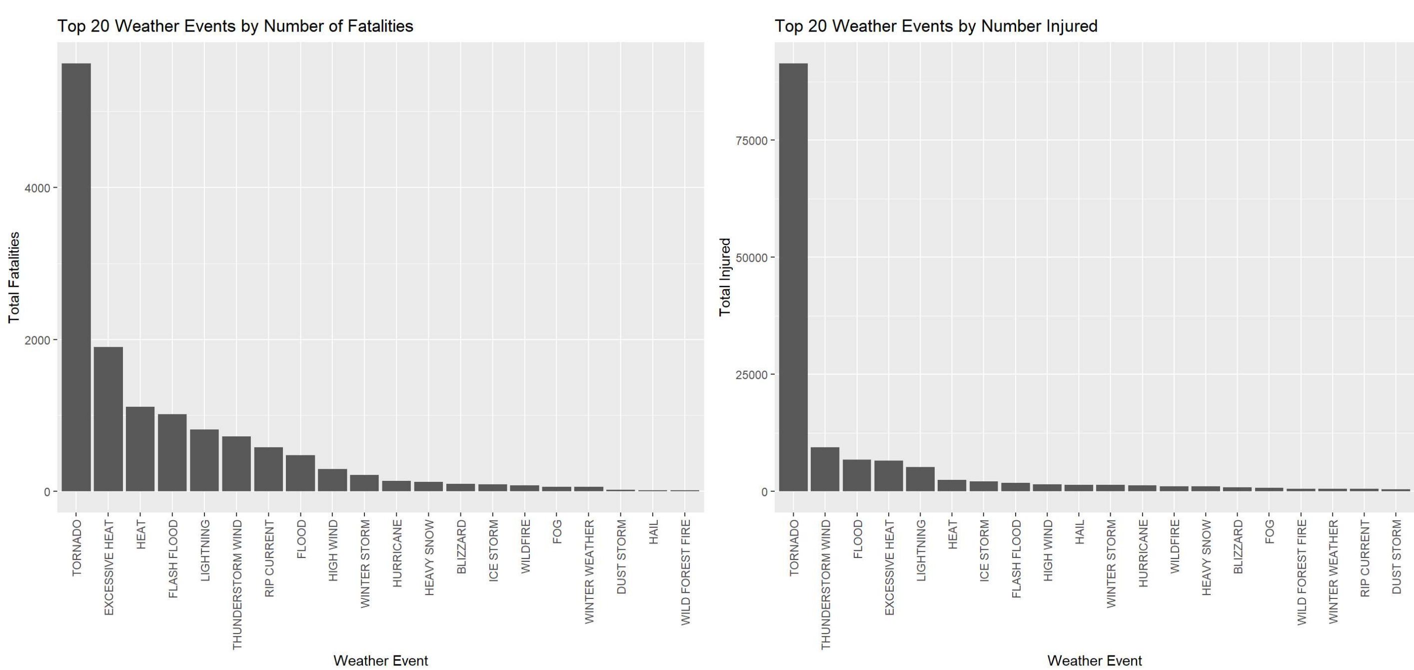
Showing 1 to 10 of 749 entries Previous 1 2 3 4 5 … 75 Next

Original EVTYPE(unique_events) mapped to the event grouping as created by semantic and string distance clustering

Results

The primary danger in terms of fatalities and injuries to the population is tornado events. The heat/excessive heat should be of high concern as well since they are the 2nd most common causes of fatalities.

```
grid.arrange(fatal_plot, injury_plot, ncol=2)
```



The top cause of property damage is tornadoes closely followed by thunderstorms with accompanying winds. Crop damage is primarily caused by hail with roughly triple the amount of damage of the next most common cause.

```
grid.arrange(prop_damage_plot, crop_damage_plot, ncol=2)
```