**Supplementary Table 1: Pubmed example search strategy**

|  |  |
| --- | --- |
| Search term validation | The food assistance search terms were taken from Martin et al. The machine learning terms were taken from Wang et al.  |
| #1 | "Food Assistance"[Mesh] OR "Food Supply"[Mesh] |
| #2 | (food\*[tw] OR meal\*[tw] OR nutrition[tw] OR dietary[tw] OR diet[tw] OR "infant formula"[tw] OR "baby formula"[tw]) AND (access\*[tw] OR suppl\*[tw] OR assist\*[tw] OR aid[tw] OR secure[tw] OR securit\*[tw] OR insecure[tw] OR insecurit\*[tw] OR pantr\*[tw] OR bank\*[tw] OR distribut\*[tw] OR desert\*[tw] OR system\*[tw] OR relief[tw] OR rescue\*[tw] OR redistribut\*[tw] OR response\*[tw] OR responder\*[tw] OR resilienc\*[tw]) |
| #3 | foodbank\*[tw] OR SNAP[tw] OR "food stamp\*"[tw] OR "Women Infants and Children Program"[tw] OR WIC[tw] OR "food first responder\*"[tw] OR "feed the hungry"[tw] OR "emergency kitchen\*"[tw] OR "free meal\*"[tw] OR "free food\*"[tw] OR "community pantr\*"[tw] OR "food service\*"[tw] OR "meal service\*"[tw] OR "soup kitchen\*"[tw] OR "community meal\*"[tw] |
| #4 | #1 or #2 or #3 |
| #5 | ("Machine Learning"[Mesh] OR "Artificial Intelligence"[Mesh] OR "Natural Language Processing"[Mesh] OR "Neural Networks(Computer)"[Mesh] OR "Support Vector Machine"[Mesh] OR Machine learning[Title/Abstract] OR Artificial Intelligence[Title/Abstract] OR Naive Bayes[Title/Abstract] OR bayesian learning[Title/Abstract] OR Neural network[Title/Abstract] OR Neural networks[Title/Abstract] OR Natural language processing[Title/Abstract] OR support vector\*[Title/Abstract] OR random forest\*[Title/Abstract] OR boosting[Title/Abstract] OR deep learning[Title/Abstract] OR machine intelligence[Title/Abstract] OR computational intelligence[Title/Abstract] OR computer reasoning[Title/Abstract]) |
| #6 | ((sentiment[Title/Abstract] OR sentiments[Title] OR opinion[Title] OR opinions[Title] OR emotion[Title] OR emotions[Title] OR emotive[Title] OR affect[Title] OR affects[Title] OR affective[Title]) AND (“sentiment classification” OR “opinion mining” OR “natural language processing” OR NLP OR “text analytics” OR “text mining” OR “F-measure” OR “emotion classification”)) OR “sentiment analysis” |
| #7 | (“social network analysis”[All Fields] OR “network analysis”[All Fields]) OR “social media network analysis” OR “Twitter”) |
| #8 | #5 or # 6 or #7 |
| #12 | #4 and #8  |
| #13 | Date Limit: 2010 to Current |

Supplementary Figure 1: **Model hyperparameter selection of topic number, alpha, and beta**



Caption: In order to select the optimal number of topics to analyze from the corpus, the coherence score, alpha, and beta model hyperparameters were calculated on 75% of the corpus as described by Kapadia.61 Using Matplotlib99 package in Python, coherence values are plotted on the y axis versus number of topics on the x axis. 9 topics had the highest coherence score.

**Supplementary Figure 2: Intertopic distance map (via multidimensional scaling)**



Caption: The pyLDAvis62 algorithm package in Python was used to produce a dynamic intertopic distance map (via multidimensional scaling). Each circle represents a topic. The size of the circle relates to that topic’s relative dominance within the corpus. The ideal map has little overlap between each topic. This map acts, therefore, as a visual check of the coherence value and topic number presented in Supplementary Figure 1.

**Supplementary Table 2: Results of LDA model: Topics with percent contribution**

|  |  |  |
| --- | --- | --- |
| # | Topics | Percent Contribution |
| 1 | 0.037\*"churn" + 0.015\*"client" + 0.015\*"benefit" + 0.014\*"state" + ' '0.012\*"cost" + 0.012\*"government" + 0.010\*"recertification" + ' '0.010\*"program" + 0.009\*"agency" + 0.008\*"fraud"' | 0.6359 |
| 2 | 0.020\*"snap" + 0.019\*"household" + 0.009\*"new" + 0.009\*"recipient" + ''0.007\*"topic" + 0.007\*"program" + 0.006\*"exist" + 0.006\*"insecurity" + ''0.006\*"include" + 0.005\*"level"' | 0.9996 |
| 3 | 0.011\*"amount" + 0.010\*"supermarket" + 0.009\*"receive" + 0.008\*"bank" + ''0.006\*"warehouse" + 0.006\*"layer" + 0.006\*"event" + 0.005\*"method" + ''0.005\*"approximation" + 0.005\*"regional"' | NA |
| 4 | 0.000\*"security" + 0.000\*"blockchain" + 0.000\*"level" + 0.000\*"measure" + ''0.000\*"household" + 0.000\*"insecurity" + 0.000\*"drone" + 0.000\*"grateful" + ''0.000\*"coordinator" + 0.000\*"relentlessly"' | NA |
| 5 | 0.034\*"tweet" + 0.016\*"classifier" + 0.011\*"sentiment" + 0.010\*"word" + ''0.009\*"feature" + 0.007\*"trend" + 0.006\*"propose" + 0.006\*"dataset" + ''0.006\*"negative" + 0.006\*"positive"' | 0.9995 |
| 6 | 0.020\*"demand" + 0.014\*"change" + 0.011\*"country" + 0.011\*"product" + ''0.007\*"increase" + 0.007\*"structure" + 0.007\*"consumer" + 0.005\*"value" + ''0.005\*"pandemic" + 0.004\*"flour"' | 0.9087 |
| 7 |  '0.010\*"system" + 0.009\*"rescue" + 0.007\*"value" + 0.007\*"volunteer" + ''0.007\*"network" + 0.007\*"donation" + 0.006\*"forecast" + 0.006\*"give" + ''0.005\*"problem" + 0.005\*"expert"' | 0.9996 |
| 8 | 0.024\*"security" + 0.013\*"household" + 0.007\*"level" + 0.007\*"measure" + ' '0.006\*"class" + 0.006\*"crisis" + 0.006\*"insecurity" + 0.005\*"country" + ' '0.005\*"access" + 0.005\*"prediction"' | 0.9996 |
| 9 | '0.016\*"mention" + 0.010\*"hurricane" + 0.008\*"twitter" + 0.008\*"healthy" + ' '0.008\*"tweet" + 0.006\*"category" + 0.006\*"list" + 0.006\*"day" + ' '0.006\*"group" + 0.006\*"examine"' | 0.9995 |

**Supplementary Figure 3: Importance of topic key words: representative dominant (Topics 1, 2, 7) and non-dominant (Topic 3) topics**



Caption: The frequency and density of keywords in the topics were calculated using TextBlob100 package in Python, shown in **Supplementary Figure 3**. These graphs show word count (left axes) versus word density (right axes) for three representative dominant topics (topics 1, 2, and 7) and one representative non-dominant topic (topic 3). Words that had lower word density than word count across multiple topics were removed from the analysis. This included, for example, *food*, *data*, and *model*.

**Supplementary Figure 4: Top 10 word frequencies related to disasters and emergencies**



Using Textblob package in Python, word frequencies related to disasters and emergencies were calculated. 100 The top ten words and their word counts were plotted in a bar graph using Matplotlib99 package in Python.