A.1 Data

A.1.1 Data Collection

We collected Twitter data beginning in 2012. However, the tweets collected during 2012-13 flu season were removed in this study, because the data did not cover the complete flu season. We discarded retweets and non-English tweets.[1] For the CDC data, we collected the data from the 2013 to 2017 flu seasons, where each flu season starts in July and ends in May in the following year. To match CDC data, we removed tweets posted in June. The statistical description of our final data is listed in Table 1.

Table 1. Overview of Twitter data in this study

Flu Season	Tweet count	Unique user count
2013 July - 2014 May	264,171	199,733
2014 July - 2015 May	336,644	219,012
2015 July - 2016 May	232,591	147,564
2016 July - 2017 May	263,535	175,770
Total	1,124,839	742,079

A.1.2 Data Preprocessing

Tweets have some unique characteristics that do not exist in traditional text, such as hashtags, hyperlinks, and colloquial language. To make the text more appropriate for natural language processing tools, we preprocessed each tweet according to the following steps:

1. Hyperlinks, hashtags, user mentions in each tweet were replaced with “<url>”, “<hashtag>”, and “<user>” respectively.
2. Repeated punctuation was replaced with “[punctuation] <repeat>”.
3. Each tweet was lowercased and tokenized using NLTK.[2]

A.1.3 Data Annotation

To build training data, we collected annotations for a random sample of 10,000 tweets from the full collection. Annotations were obtained from Amazon Mechanical Turk,[3] with three independent annotations per tweet. Tweets were labeled with the following:

- Does this message indicate that someone received, or intended to receive, a flu vaccine? (yes or no)
    - If yes: has the person already received a vaccine, or do they intend to receive the vaccine in the future.

We refer to tweets labeled “yes” as “intention/receipt” and tweets labeled “no” as “other”.

We rejected annotators whose agreement was anomalously low (percentage agreement was ≤ 60%). Three bad annotators were removed from our final dataset. We took a majority vote on the remaining 29,970 annotations to obtain the final labels. If there was not a majority label, then we defaulted to the "other" label. The dataset contained 10,000 tweets, with 32.8% labeled as positive for "intention/receipt", with a kappa score of 0.79, using Fleiss' kappa to measure the inter-annotator agreement.[4] Then we manually corrected 168 labels of the dataset and finally obtained 31.1% labeled as positive for "intention/receipt".

A.2 Automatic Assessment Methods

To automatically identify tweets expressing vaccination intention/receipt, we used the labeled data to train two machine learning classifiers: Logistic Regression (LR) and Convolutional Neural Network (CNN). The LR model achieved the best performance among other classifiers in our previous study.[5] We implemented Logistic Regression (LR) classifier using the scikit-learn toolkit.[6] CNN has been drawn significant attention in recent years because of its impressive performance on text classification tasks.[7] We trained the two models on the annotated Twitter data. After optimizing the model parameters and hyperparameters, we compared the two models. We finally chose the model that achieved the best performance in the validation experiments.

A.2.1 Logistic Regression

We fed the LR model with TF-IDF weighted n-gram (uni-, bi- and tri-gram) features, as well as part-of-speech (POS) counts from TweeboParser,[8] and emoji and emoticon features derived from two open lexicons.[9, 10] Feature counts were normalized to sum to 1 within each tweet. The list of features we used in this study are shown in Table 2.

Table 2 Details of the feature set for Logistic Regression classifier

Feature name	Feature attributes
N-gram	TF-IDF scores of unigrams, bigrams, trigrams
Part-of-Speech	Counts of POS tags, normalized by the total tags in the tweet
Emoji	Counts of negative and positive emojis, normalized by total counts.
Emoticon	Counts of negative and positive emoticons, normalized by total counts.

We balanced the weight of each label by adjusting weights inversely proportional to class frequencies in the training dataset. We adopted cross entropy as the loss function with $l2$norm penalty for weight regularization.

A.2.2 Convolutional Neural Network

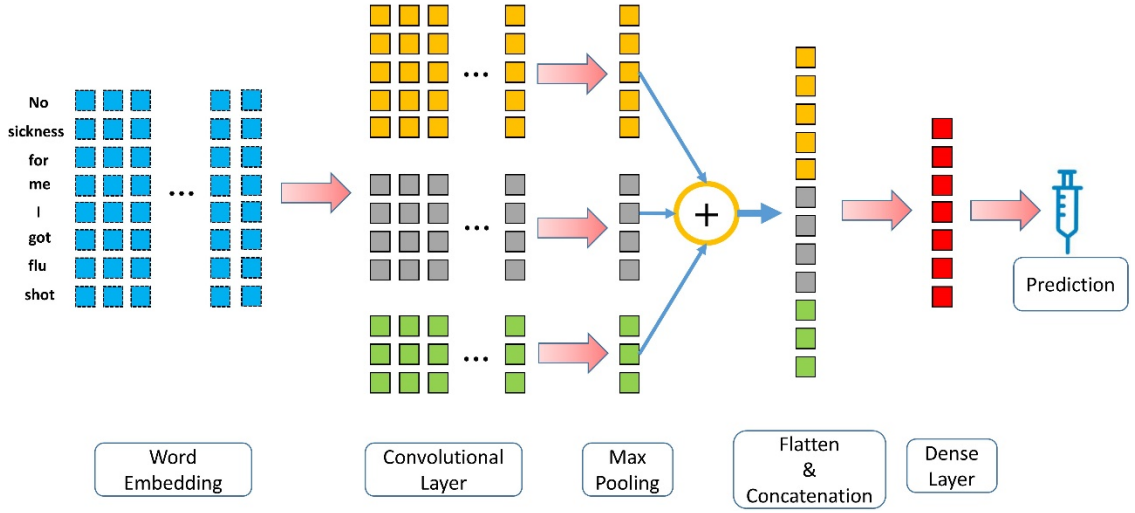


Figure 1. The architecture of the CNN model.

The embedding layer converts processed tweets into an embedding matrix of floating point values, where each row is a vector representation of a word. The embedding matrix is then fed into the convolutional layer, where the matrix will be screened and sampled by the filters. We set 150 filters in this layer. Each filter is a square sliding window and we defined three different sizes of filters: 3*3, 4*4, 5*5. We set the filter stride to 1 and padding mode to "VALID". We obtained the squares by sliding the filters over the matrix. Those captured squares will be fed into the next layer, the pooling layer. We adopt 1-max pooling as the strategy to extract a max scalar value from each square, which outputs the maximum value. We stack another convolutional layer and pooling layer following the first pooling layer, for which the operation steps are the same.

Outputs from the stacked convolutional and pooling layers are flattened, concatenated and fed to the next layer, the dense layer, where it learns and generates a fixed representation for each tweet. We set the activation function as rectified linear unit (ReLU).[11] We set the output dimension of this dense layer to 150. A dropout was applied in the layer, where dropout is a standard method to prevent overfitting by randomly set a proportion of values to zero during training.[12]

We fed the outputs from the dense layer to the sigmoid function to predict the final binary label, "intention/receipt" or "other". We adopted the binary cross entropy function with $l2$ penalty to calculate the loss of predictions. Adam with a learning rate of 0.001 and decay of 0.003 was adopted to optimize the parameters.[13]

A.2.3 Experiment Settings

We randomly sliced the dataset into three pieces: 80% as training set, 10% as development set and 10% as testing set. We trained our two methods, LR and CNN, on the training set, tuned parameters on the development set, and evaluated the methods on the testing set. We

balanced weights of predicted labels in the two models. The models' parameters were selected by accuracy on the development set. The CNN model was trained by 10 epochs, batch size was set by 64, and the dropout rate was set to 0.2. We fixed the length of inputs by either padding sentence to 40 words or slicing the first 40 words. Outputs of the classifiers are probabilities of "intention/receipt", which consider true only if the values are equal to or larger than 0.5 and vice versa. "Precision", "recall", "f1-score" were used to evaluate the performance of each method on the testing set. We focused on the performance of "intention/receipt", not "other" label, which consistently keeps the same evaluation metrics with our previous work.[5]

A.2.4 Selecting Word Embeddings

Word embedding is a language modeling technique that maps words into a set of word vectors.[14] The CNN model in our study was fed with the word vectors. There are two popular frameworks to generate the vectors, Word2vec and GloVe.[14, 15] We selected the best embedding model from the following options:

1. We obtained pre-trained word embedding by running word2vec from Gensim over our collected tweet dataset.[16] We set the tool's default settings except for changing minimum count of words to 1 and number of iterations to 15. We finally obtained 100 dimensional embedding for each word (denoted as *word2vec*).
2. We obtained an embedding model by GloVe with its default parameter settings from its official website (denoted as *glovec*).
3. Google provides pretrained word2vec embeddings on its news dataset,[14] and Stanford provides pretrained GloVe embeddings on its Twitter dataset (denoted as *pre-word2vec* and *pre-glovec* respectively).[15]
4. Character-level embeddings have recently been shown to perform well on text classification.[17] We built word embeddings using a one-hot encoding of characters (denoted as *character*).

We fed the different embedding models to the same CNN model with the fixed parameters. We evaluated the performance by precision, recall and F1-score. The performance is shown in Table 3.

Table 3 Performance of different word embeddings on our dataset.

Word Embeddings	Precision	Recall	F1-score
word2vec	0.894	0.800	**0.843**
glovec	0.820	0.751	0.784
pre-glovec	0.794	0.800	0.797
pre-word2vec	0.895	0.767	0.826
character	0.858	0.729	0.788

Finally, we chose the *word2vec* model trained on the collected data in this study, because it achieves the best performance. We also trained embeddings with 50 and 200 dimensions for both Word2vec and GloVe, but their performance was worse than with 100 dimensions. The word embedding trained on our collected data outperformed pre-trained models from Google and Stanford. Thus, we chose this embedding model for our experiments.

A.2.5 Test Performance of Classifiers

Table 4 Classification performance on test data.

Method	Precision	Recall	F1-score
LR-ngram*	0.837	0.799	0.818
CNN-embedding	0.894	0.800	**0.843**
LR-embedding-average	0.828	0.651	0.729

We used the precision, recall, and F1-score to measure the performance of the two classifiers. We selected the classifier for our analysis tasks based on the best F1-score. We show the test performance in Table 4, where embedding refers to the word vectors from the selected word2vec model, and embedding-average means the trained features of LR are word vectors created by averaging the word vectors of all words in each tweet. Compared to the other two models, the CNN-embedding has better precision and F1-score. We finally selected CNN-embedding for categorizing all the tweets we collected.

A.3 Validation Experiments

In this section, we provide additional details and experiments on the validation process of comparing the Twitter data to the CDC data.

A.3.1 Experimental Steps

We ran both classifiers (LR and CNN) on all tweets from the 2013 to 2017 seasons to obtain labeled tweets. We restricted the analysis to tweets from the United States. We validated our approach across three dimensions: time, geography, and demography.

- Time:
    a. We counted both the weekly and monthly number of tweets classified as “intention/receipt”. To be consistent with CDC’s week definitions, we used the epidemiological week instead of the ISO week to calculate the counts. The data from Twitter and CDC were normalized by z-score separately.
    b. Because the types of data were time-series, we ran the time series model, “autoregressive integrated moving average” (ARIMA), to obtain relationship Twitter and CDC, which was (p, d, q) = (0, 1, 0). The result suggested a linear

relationship between the trends of CDC and Twitter. We then fitted the time series data by a linear regression model using Twitter trends to predict CDC trends.

c. We additionally calculated Pearson correlation and Spearman correlation scores on the Twitter counts and CDC data.

- Geography:
    a. For geographic regions (referred to as “Region”), we aggregated the total counts of “intention/receipt” tweets for the 10 HHS regions separately. In the “Region-year” experiment, we treated the regional tweets in each flu season as a separate point. We normalized the counts of “Region” and “Region-year” by dividing the number of tweets from that region, using the random sample of tweets from the Twitter streaming API.
    b. For “State” and “State-year”, we excluded five locations, Northern Mariana Islands, US Virgin Islands, Puerto Rico, Guam, and District of Columbia. These experiments follow the same process as the region experiments, but within individual US states.
    c. All the values were normalized by z-scores.
    d. We validated the geographic data by measuring Pearson and Spearman correlations.
- Demography:
    a. For “Gender”, we first counted positive tweets separately for males and females for each flu season. We divided the female counts by male counts of each flu season to generate gender ratios for the Twitter data. Finally, the ratios were normalized by z-score.

A.3.2 Correlation Results

Table 5.1 shows the Pearson correlations over time for both the CNN and LR models. Table 5.2 shows the correlations over geography for the LR model.

Table 5.1 Validation by Pearson correlation for time.

Validation model	All	2013-14 season	2014-15 season	2015-16 season	2016-17 season
CNN	0.899	0.897	0.985	0.985	0.967
LR	0.897	0.927	0.992	0.985	0.984

Table 5.2 Validation of LR by Pearson correlation for geography.

Validation model	State	State year	Region	Region year
LR	0.433	0.212	0.456	-0.121

Table 6.1 shows the Spearman correlation by time, and Table 6.2 shows the Spearman correlation by geography.

As the data is split into finer granularities, such as State or State-year, the correlation scores tend to decrease. This might be caused by a smaller sample size of tweets in the smaller bins. This suggests that if we could obtain more data, this approach will be more accurate.

Table 6.1 Validation by Spearman correlation for time.

Validation model	All	2013-14 season	2014-15 season	2015-16 season	2016-17 season
CNN	0.929	0.948	0.970	0.900	0.943
LR	0.934	0.957	0.975	0.936	0.943

Table 6.2 Validation by Spearman correlation score for geography.

Validation model	State	State year	Region	Region year
CNN	0.402	0.236	0.552	-0.088
LR	0.446	0.208	0.455	-0.133

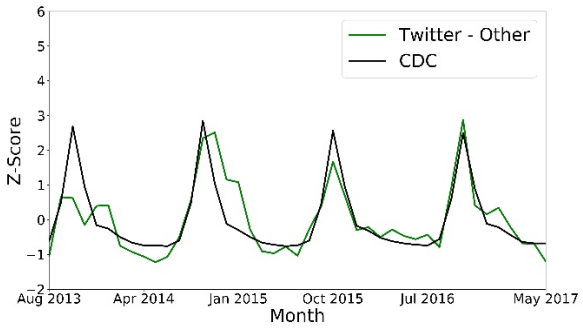
A.3.4 Validation of “Other” Tweets

We have focused on the “intention/receipt” tweets under the assumption that they will be more meaningful than the tweets classified as “other”, i.e., tweets that contain vaccine-related phrases but do not explicitly state that someone received or intends to receive a vaccine. In this section, we measured the predictive value of the “other” tweets, which might also correlate with CDC data, and we compare the correlations to the correlations of the “intention/receipt” tweets.

We kept the same experiment settings for the tweets of the “other” label as the “intention/receipt” tweets. We calculated the Pearson correlation with the CDC data. The results are shown in Table 7. We plot the monthly flu vaccine prevalence between “other” (denote as Twitter-Other) and the CDC and weekly prevalence of Twitter data in Figure 2. The “other” tweets have lower Pearson correlation than “intention/receipt” tweets overall with the CDC data. In Figure 2.2, the other tweets in the dataset have very high week-to-week variability, with numerous spikes that do not fit the seasonal trends. This suggests that our classifier is reducing the noise and improving our identification of vaccine behaviors.

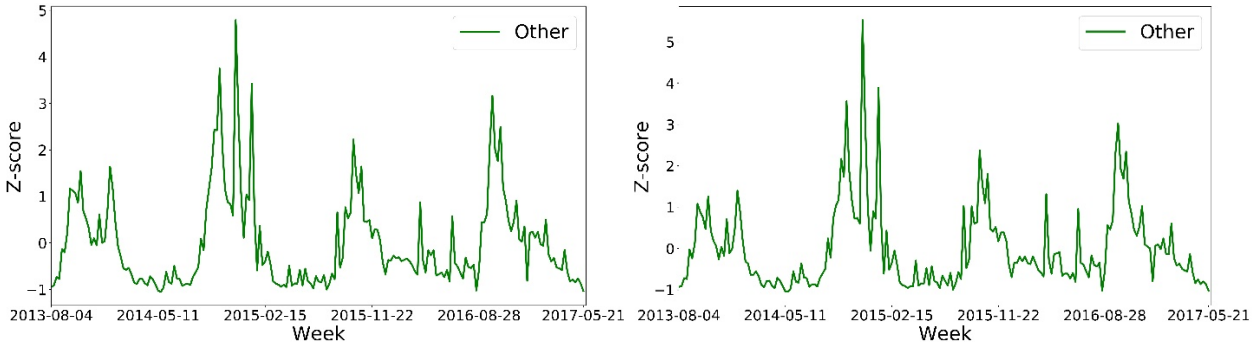
Table 7 Validation Results of CNN and LR by “other” label.

Validation Task	CNN	LR
All seasons	0.820	0.844
State	0.173	0.200
State-year	0.111	0.134
Region	0.587	0.589
Region-year	0.451	0.500



(a) LR (b) CNN

Figure 2.1 Monthly prevalence of “Other” trends from Twitter compared to the CDC.

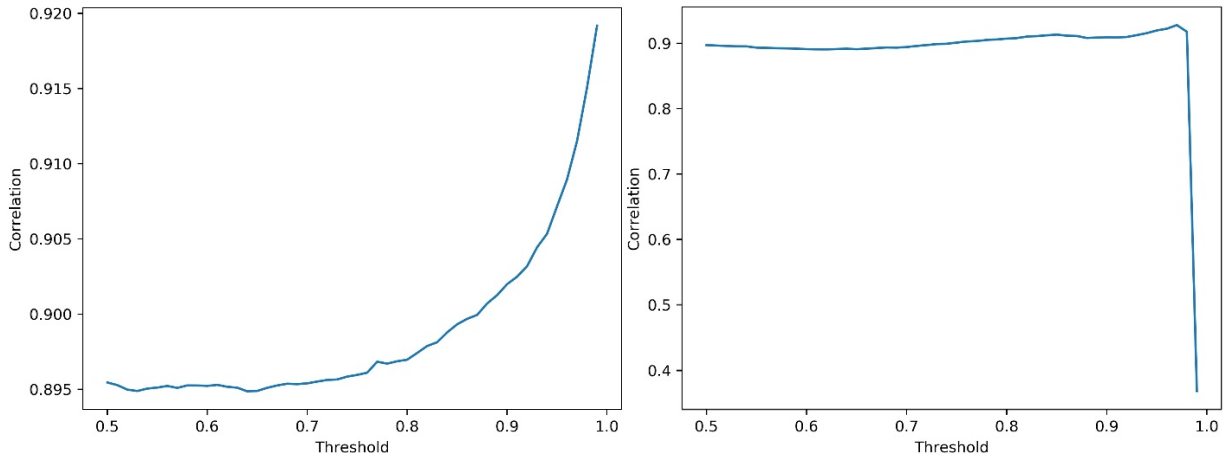


(a) LR (b) CNN

Figure 2.2. Weekly time series of tweets classified as “Other” by LR (a) and CNN (b).

A.4 Additional Analyses

A.4.1 Sensitivity of the Classification Threshold

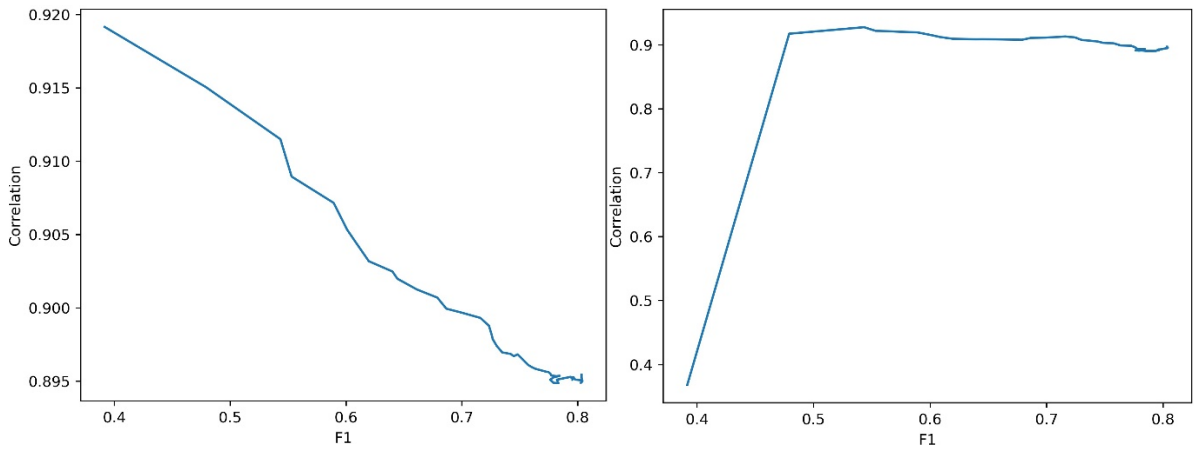


(a) LR (b) CNN

Figure 3. The relationship between the prediction threshold and correlation coefficient.

In this section, we explore how the threshold of classifiers impacts the Pearson correlation. Specifically, the threshold of how the probability of a tweet being positive before it is actually positive. By default, anything with probability greater than or equal to 0.5 is classified as positive, but this threshold can be raised to increase precision (at the expense of recall).

In Figure 3(a) and 3(b), we plotted the relationship between Pearson correlation and prediction threshold for both LR and CNN. Both approaches show that increasing the predicting threshold can improve the correlation coefficient. Increasing the threshold indicates higher confidence of the classifier, that is to say, a tweet will only be considered as “intention/receipt” when the classifier has high confidence. In the view of the classifier, only the tweets have enough evidence to indicate vaccination will be classified as “intention/receipt”. Additionally, we could find that when the threshold of CNN is set to near 0.950, the correlation score decreases rapidly, so raising the threshold does not always improve performance monotonically.



(a) LR (b) CNN

Figure 4 The relationship between the F1 score and correlation coefficient.

.

In Figure 4(a) and 4(b), we explore the relationship between the F1-score and Pearson correlation, because our criteria for selecting the best classifier was by F1-score. The CNN classifier reaches the highest correlation coefficient at around an F1-score of 0.500. Under both models, the correlation drops when the F1 score is too high, likely because the optimal balance is high precision and low recall, even if that drops the F1 score.

For the LR model, while the correlation varies with F1 score, the correlation values are all very similar, and all are above .900. However, the CNN model is not very stable with respect to the correlation coefficient, which might indicate the LR is more robust. We also combined the two approaches to see if we could achieve better performance in the next section.

A.4.2 An Ensemble Perspective of the Two Models

We combined the two models using two linear combination approaches: combining monthly counts of tweets from the LR and CNN (weighted-counts), and combining the prediction probabilities of each approach (weighted-prob). We calculated the combination by the formulas below:

$$Weighted - output = \sum_{i=1}^{2} W_i * X_i \qquad (1),$$

$$W_i = \frac{F1_i}{\sum_{i=1}^{2} F1_i} \qquad (2),$$

where F1 is the F1-score of each classifier achieved on the test data, and X_i is the count number of each classifier for “weighted-counts” or the predicted probability of “intention/receipt” of each tweet by i-th classifier. Specifically, the weighted-count is the weighted sum of weighted counts from the LR and CNN approaches; for weighted-prob, instead of counts, we calculated the prediction probability of each tweet by the weighted sum of the probabilities from each classifier. The F1-score of each method was used as the weight in the Equation (1). The weights were normalized by the sum of weights to ensure they are within 0 and 1, as shown in Equation (2).

For the validation, we evaluated the performance of the tweets classified as “intention/receipt” and “other”. We validated the two ensemble approaches by calculating Pearson correlation with the CDC data. The results are shown in Table 8. We find that the weighted-counts performs slightly better than the weighted-prob on the tweets classified as “intention/receipt”. The ensemble ways show promising results, outperforming a single classifier.

Table 8. Validation Results of CNN and LR.

Validation Task	Intention/receipt		Other	
	Weighted-Counts	Weighted-Prob	Weighted-Counts	Weighted-Prob
All seasons	0.899	0.895	0.835	0.840
State	0.406	0.437	0.188	0.192
State-year	0.296	0.281	0.092	0.115
Region	0.475	0.432	0.588	0.591
Region-year	0.325	0.264	0.480	0.497

A.4.3 Simpson’s Paradox

In our previous work,[5] LR achieved a .90 correlation on the three consecutive flu seasons (2013-14, 2014-15, 2015-16). In this work, we added a fourth flu season, and LR received a lower correlation score after adding the 2016-17 season. To explore why the correlation dropped, we calculated the correlation on the 2016-17 by itself, to see if this season had a lower correlation that caused the overall correlation to drop. The results are shown in Table 9, comparing the first three seasons (2013-16), the fourth season (2016-17), and all four seasons.

Surprisingly, we discovered that the CNN achieves lower correlation scores than LR on both Seasons 2013-16 and Season 2016-17, even though it exceeds LR on all seasons. This behavior could be explained by “Simpson’s paradox”, a common paradoxical phenomenon in data analysis.[18]

Table 9 Pearson correlation of two different time periods.

Validation Task	Intention/receipt	
	CNN	LR
Seasons 2013-16	0.892	**0.903**
Season 2016-17	0.967	**0.984**
All seasons	**0.899**	0.897

A.4.4 Additional Trend Figures

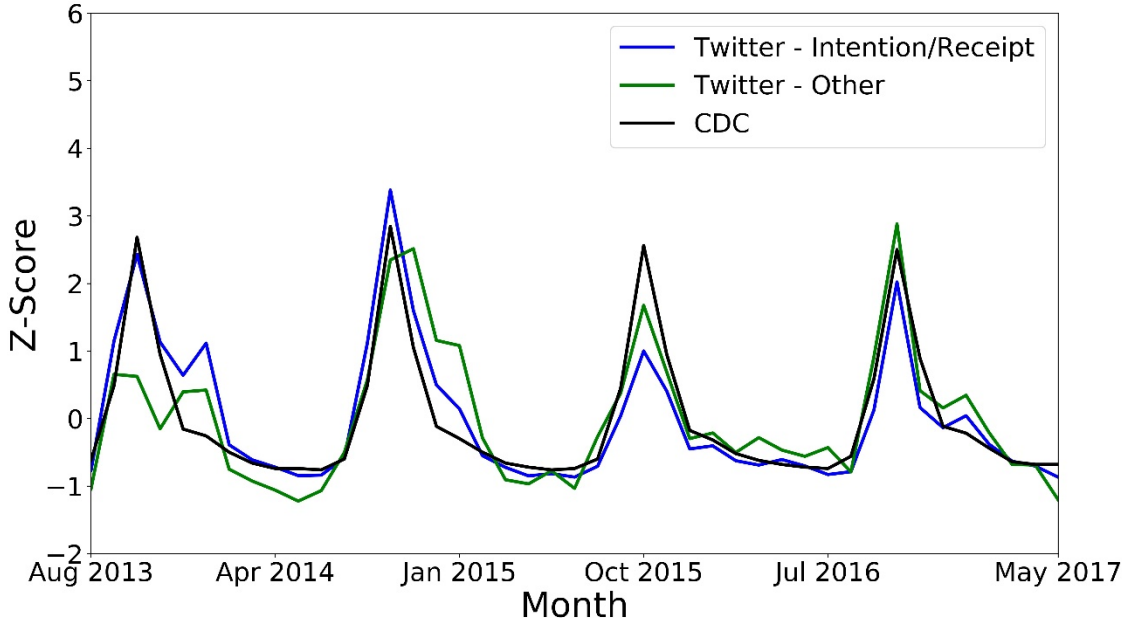
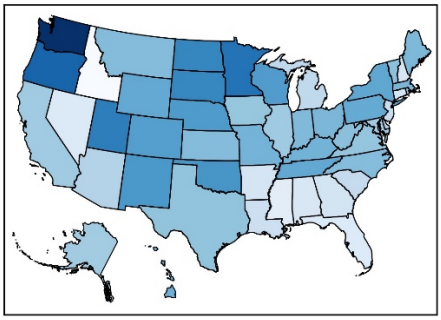
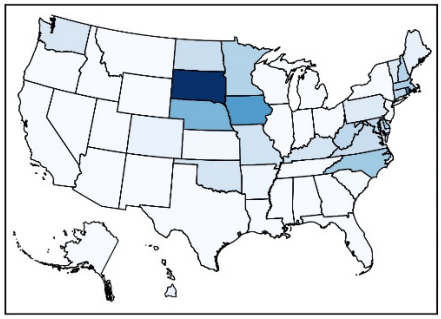
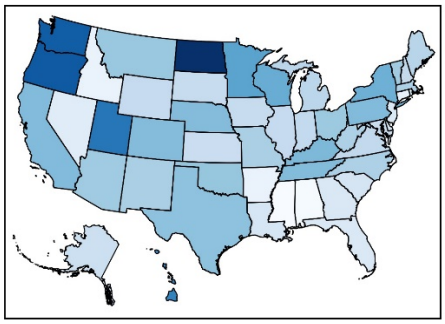
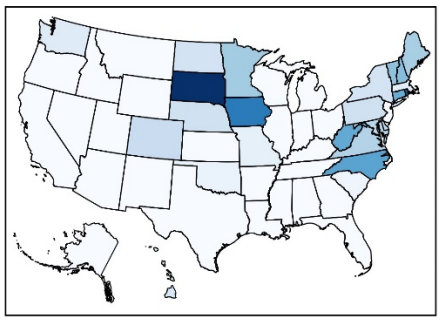
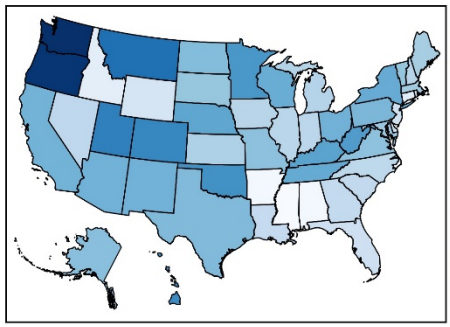
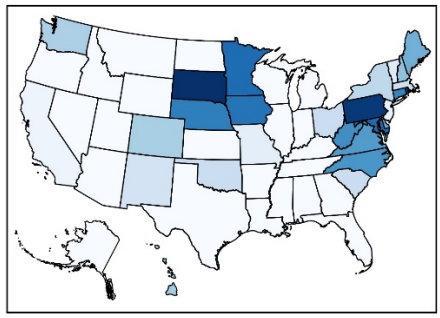
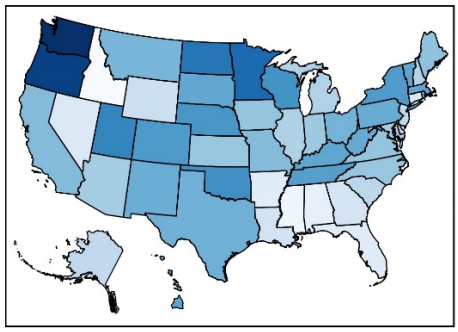
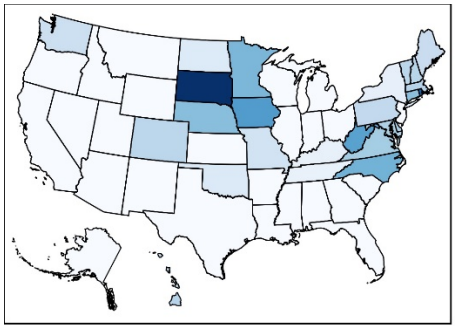


Figure 5. Monthly prevalence of vaccination trends from Twitter and CDC.

Figure 5 shows both the CNN time series (blue) alongside the LR time series (green) and CDC data. There are only minor differences in the trends of the two models. Notice that each peak of the plots is usually in October of the flu season. Yet, there is a distinct peak between Jan. 2014 and Feb. 2014, which might indicate many people also talked about taking flu vaccination shots during that time.

We visualized vaccine coverage in the 50 states each flu season in the Figure 6.[19] We find there are some similar patterns between the Twitter and CDC that the states in the northeast of US show high vaccine coverage and southeast of the US show the lower vaccine coverage, while there are also some clear differences, for example, in the Twitter data, Washington and Oregon show consistently very dark colors.

Figure 6. Flu vaccine trends of both the Twitter and CDC in the U.S.

Flu Season	Twitter	CDC
2014-15		
2015-16		
2016-17		
All		

References

1. Lui M. saffsd/langid.py. GitHub.
2. Bird S, Loper E. NLTK: the natural language toolkit. In: *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics; 2004:31. doi:10.3115/1219044.1219075
3. Callison-Burch C, Dredze M. Creating speech and language data with Amazon's Mechanical Turk. In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. ; 2010:1-12.
4. Fleiss JL. Measuring nominal scale agreement among many raters. *Psychol Bull*. 1971;76(5):378-382. doi:10.1037/h0031619
5. Huang X, Smith MC, Paul M, et al. Examining patterns of influenza vaccination in social media. In: *AAAI Joint Workshop on Health Intelligence (W3PHIAI)*. ; 2017:542-546.
6. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python. *J Mach Learn Res*. 2011;12(Oct):2825-2830.
7. Kim Y. Convolutional Neural Networks for Sentence Classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, {EMNLP} 2014, October 25-29, 2014, Doha, Qatar, {A} Meeting of SIGDAT, a Special Interest Group of the {ACL}*. ; 2014:1746-1751.
8. Gimpel K, Schneider N, O'Connor B, et al. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics; 2011:42-47.
9. Kralj Novak P, Smailović J, Sluban B, et al. Sentiment of Emojis. *PLoS One*. 2015;10(12):e0144296. doi:10.1371/journal.pone.0144296
10. Mohammad SM, Turney PD. Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. CAAGET '10. Stroudsburg, PA, USA: Association for Computational Linguistics; 2010:26-34.
11. Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. USA: Omnipress; 2010:807-814.
12. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J Mach Learn Res*. 2014;15:1929-1958.
13. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv Prepr arXiv14126980*. 2014.
14. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Lake Tahoe, Nevada: Curran Associates Inc.; 2013:3111-3119.
15. Pennington J, Socher R, Manning CD. GloVe: Global Vectors for Word Representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. ; 2014:1532-1543. http://www.aclweb.org/anthology/D14-1162.
16. Rehurek R, Sojka P. Software framework for topic modelling with large corpora. In: *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ; 2010:45-50.
17. Kim Y, Jernite Y, Sontag D, et al.. Character-Aware Neural Language Models. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. Phoenix, Arizona;

2016:2741-2749.
18. Pearl J. Comment: Understanding Simpson's Paradox. *Am Stat*. 2014;68(1):8-13. doi:10.1080/00031305.2014.876829
19. Root B. matplotlib/basemap. GitHub.