Learning COVID-19 Risk Factors with LSTMs

GROUP 2: Blaine Rothrock, Grant Gasser, David Terpay

Abstract

We present a method to understand and identify under-reported risk factors related to the novel Coronavirus (COVID-19). First, we manually identify risk factors as reported by the CDC, setting aside a holdout set to test whether those risk factors can be identified by the model. We define under-reported risk factors as those considered risk factors by our model but not being found on the CDC's website. The remainder of our approach consists of three primary steps. The first consists of training a deep language model on the CORD-19 data set (Wang et al., 2020). The second is another pass through the training corpus, storing the hidden state h_t for each token where we're particularly interested in the hidden states of the risk factors. In the third pass, we compute similarities of tokens to risk factors using a token's h_t compared to the average hidden state of each of the provided risk factors. We then compare the risk factors in the holdout set to see if they are similar to the provided risk factors. Although our method struggles to reliably identify holdout tokens as risk factors, we show that the model is able to learn that the holdout risk factor tokens are more similar to the provided risk factors than non-holdout tokens, proving that the model achieves some intuition about COVID-19 risk factors.

1 Introduction

There is great uncertainty surrounding COVID-19 and a flood of information from research institutions surrounding the virus. Much of what is known is due to the urgency of researchers worldwide to find a stable vaccine. As such, information reported about how to protect oneself from the disease is constantly changing. We determined that it would be informative to understand how and why existing risk factors are related and then use this information to detect under-reported risk factors.

Given the current global context, this type of research is very valuable as it can highlight how risk factors are discussed in academic publications versus media outlets. Any differences could inform us of risk factors we should be looking out for but are not. Additionally, the embeddings for the risk factors could be used to further research COVID-19 risk factors.

As there is no research available on this topic, our benchmark consists of evaluating how well our model extracted risk factors relative to the manually defined risk factors. We evaluate our language model using perplexity for training confirmation. Our process was first tested using special characters, punctuation and single-digit integers on the WikiText-2 data set (Merity et al., 2016) to ensure some level of functionality before producing results on a corpus created using the CORD-19 data set those results are omitted from this text.

2 Related Work

Learning higher dimensional vector representations of token in the training corpus is the main goal of a language model. It has been shown that this vector space can hold contextual information about the tokens themselves (Mikolov et al., 2013). Hidden-states, often represented in the same dimensional as embeddings spaces are internal representations of a prediction point in the model. Recurrent neural networks (RNNs) store a hidden-state at each iteration as the prediction vector for the next token in a sequence (Mikolov et al., 2010). Text generation sequence-to-sequence model further utilizes a internal state as the output of the encoder layer (Sutskever et al., 2014).

3 Model

For out language model, we choose a common language model architecture based on Long Term-Short Memory (LSTM) layers (Zaremba et al., 2014). LSTMs are variate of RNN layers that are known for their ability to persist internal ”memory” using a series of logical gates and internal states. LSTMs main advantage over traditional RNNs is the logical forget gate, which allows the model to learn to mask unneeded data (Gers et al., 1999). More specifically, we chose to use bi-directional LSTMs (Ma and Hovy, 2016) because of the ability to comprehend context in both directions of a sequence. Learning new risk factors will require capturing the context surrounding a token and we not simply concern with prediction of the next token. In other words, we want as much context information as possible.

The first step of our experiment is training our language model. The model consists of embedding layer which maps a token integer representation (vocabulary index) to a higher dimensional learned vector representation. The sequence of vector representations is then fed into a series of stacked bidirectional LSTM layers with a hidden state dimension that matches the embedding dimension. Both the dimension and number of layers are configurable and discussed in the experiment section. The final output of the LSTM layers is then fed into a fully connected layer that projects the output to the size of the vocabulary. The model incorporates dropout to introduce regularization and avoid overfitting (Srivastava et al., 2014).

It is worth mentioning that LSTMs are not the state-of-the-art in language modeling research. At the time of this writing multi-headed attention based models (Vaswani et al., 2017) are the cutting edge research. While yielding lower performance, we chose LSTMs because of their ease of use and relatively small resources used to train. Our model can converge on a 10MM+ corpus (60K vocab) in under 10 hours using a consumer based GPU. We are also not necessarily interested in the predictive ability of the model, although it clearly has benefits to our evaluation. We are interested in learning accurate clustering within our embedding space. We feel, at least for initial exploration, LSTMs based architectures provide testing-bed for these experiments. Once proven reliable, perhaps transformer models can be explored. Another option is to fine-tune a model like BERT (Devlin et al., 2018), but given the large dimensional space (768) and the smaller token unit size we opted for our own approach.

4 Data and Experiments

We are using 5,000 COVID-19 research papers from the CORD-19 Open Research Data (Wang et al., 2020) set from Kaggle as the basis for our corpus. The COVID-19 Open Research Dataset (CORD-19) is a collection of research papers pulled together in collaboration with AI2, MSR, Georgetown, HIH, and The White House. This dataset has been made public on Kaggle and has been updated 25 times since it’s release. The data set contains over 134,000 scholarly articles/research papers including over 60,000 about COVID-19, SARS-CoV-2, and related coronaviruses.

The purpose of our research is to attempt to identify new risk factors in the data set using the method, described in detail in the following sections, by matching predicted tokens to a similarity of known risk factors. If our language model is fed new data and some threshold of similarity is met we can identify the next token as a potential risk factor. We evaluate this hypothesis by splitting a set of risk factors into two, one entitled ”known” and the other ”hold-out”. The hold-out set will remain unidentified and we will use it to explore similarities based on our known set.

4.1 Creating the Corpus

We select 5,000 COVID-19 articles by searching through the directory of articles and selecting an article anytime there is a match in its metadata to COVID-19 specific terminology. Next, we randomly place each article into 1 of 3 different batches for processing – training batch, test batch, and validation batch. The split that we decided to use was 80-10-10 split for training, testing, and validation respectively. Once each batch has the articles it needs, we tokenize the body of each article and insert special start and end tokens where sentences start and end. NLTK assisted in much of the needed cleaning. Next, we merge the tokens for each article into one large list of tokens for each relevant batch (training, testing, and validation). The last remaining step involves running through the batches and removing tokens with low frequency. In our case, we remove all tokens that appear less than 4 times. After tokenizing and cleaning the corpus, we have 13,095,470 tokens with a vocab size of 57,289.

4.2 Identifying Risk Factors

We found it important to find risk factors that were external to research done on this data set. Therefore, we compiled a list of 42 risk factors as identified by the CDC 1, with 12 of them being used as the holdout set 2. For the sake of simplicity, each risk factor was represented as a single token.

4.3 Training

First we trained a LSTM language model. Our model is configured for 100-dimensional embeddings and hidden state. The model consists of 3 stacked bi-directional LSTM layers (6-layers total) and a output projection layer. The output of the model log-softmax probabilities over the training vocabulary. For optimization, Negative log likelihood loss was used with the ADAM optimizer with a learning rate of 0.01 and momentum of 0.8. The model was trained over the training set with a batch size of 128 and back propagation through time length of 32. We trained for 100 epochs capturing loss and perplexity at each epoch for training, validation and test data sets. The model trained for roughly 10 hours on single Nvidia GTX-1080 GPU.

4.4 Capture Embedding and Hidden State

With a train model, we now make a pass over the data to collect embeddings and hidden states of our risk factor tokens. This is done by loading the model in inference mode (no back propagation) and iterating through the entire training set with a batch size of 1. At each iteration, we check if the current token is in out set of risk factors. If it is, we capture it's hidden state which is the final internal state of the last layer in the LSTM stack. All occurrences of the the risk factor tokens hidden states are stored and saved to disk for use in step 3.

4.5 Recording Similarity

We perform a final pass of the training data in order to capture similarity data of each token. To allow for further experimentation we simply capture similarity metrics (detailed in the next section) for every token compared to every risk factor. Using the hidden states from the previous step, we first average all to get a single 100-dimensional mean hidden state. We also do this for each of our risk factors, resulting in a mean 100-dimensional hidden state vector for each risk factor. We then iterate through the training corpus, same as in the previous sub section, and calculate the similarity to the mean of all risk factors and each individual risk factor. We save each tokens result as a row in a .csv file that can be loaded into a database for analysis via SQL.

4.6 Defining Similarity

Our first inclination was to use cosine similarity, but initial result showed similarities reaching near perfect for many tokens. We believe this is due to the number of dimensions in our hidden state. We decided to go with a variation on euclidean distance that incorporates a hyper-parameter σ that separates the tokens in order to have reasonable precision in the 100-dimensional vector space. For our experimentation we defined σ as 0.8. It is important to note the $sigma$ is directly correlated to our similarity threshold in the results section, reducing σ is the same as reducing a similarity threshold.

We define similarity between a tokens hidden state h_t and an average hidden state h as

$$sim(h_t, h) = exp(\frac{-d^2}{\sigma})$$

where d is defined as the euclidean distance between h_t and h.

5 Results

In order to identify under-reported risk factors, we compare each token to the average hidden state h of each risk factor. If $sim(h_t, h) > 0.9$ for a given token's hidden state h_t, then we count that as similar. In order to filter our 10MM tokens, we only observed tokens that were similar to at least 16 of the 30 provided risk factors and were more than 4 characters, narrowing the set of potential under-reported risk factors down to 184.

This yields a set of tokens including ”inflammation”, ”smoking”, ”pancreatic” (cancer), which could be considered risk factors. There were also tokens that may not be considered risk factors in the classic sense, but are certainly related to COVID-19 risk such as ”ventilated”, ”contact”, ”aerosolisation”, ”chroloquine”, ”Hubei”, and ”exposure”. To be fair, the vast majority of the remaining tokens were unrelated to risk factors such as ”weekend”, ”random”, ”profile”, etc. Additionally, there may be tokens that we dismissed but would be meaningful to domain experts.

In order to verify our model is able to learn the meaning of risk factors, we define a simple baseline. Since we have the similarity between each

65	elderly	nursing	facility	deficient	cancer	artery
smoking	smoked	bone	marrow	hiv	corticosteroids	congenital
obesity	bmi	diabetic	dialysis	disease	liver	pulmonary
hemoglobin	sickle	cell	thalassemia	heart	coronary	congenital
cardiomyopathies	cardiomyopathies					

Table 1: Risk Factors used for capturing model hidden states for similarity comparison against all tokens in the training corpus

old	asthma	lung	immune	deficiency	smoker	aids
obese	diabetes	kidney	cardiomyopathy	hypertension		

Table 2: Risk factors used to compare similarities. Our hypothesis is that we can identify these as Risk Factors using our similarity metric

Risk Factor	Holdout Similarity	Non-Holdout Similarity
corticosteroids	.49	.43
sickle	.48	.45
nursing	.51	.45
deficient	.49	.47
heart	.51	.46
smoking	.50	.44
liver	.51	.45
artery	.47	.46
cancer	.50	.44
hiv	.49	.42

Table 3: 10 randomly sampled risk factors and the their average similarity with holdout tokens and non-holdout tokens. Holdout similarity is higher for 10/10 risk factors. The differences are not large, but the trend is consistent.

token and the average hidden state of each risk factor, $sim(h_t, h)$, we can check if holdout risk factor tokens are more similar to the provided risk factors than non-holdout tokens. For each provided risk factor h, we average the similarity between all holdout tokens h_t and h, or Holdout Similarity:

$$\sum_{t=1}^{N} sim(h_t, h)$$

where N is the size of the holdout set. We can do the same thing with non-holdout tokens and compare both averages. If the model is learning risk factors correctly, we'd expect the average similarity for holdout tokens to be higher than non-holdout tokens, which is what we see with in Table 3.

6 Further Research/Expansion

For the sake of time and simplicity, we limited our risk factors to be one token. This forced us to break up certain risk factors such as "nursing home" into separate tokens "nursing" and "home", which clearly don't carry the same rich representation as the original term. A natural expansion of what we've done would be to examine risk factors represented by more than just one token. It would also be beneficial to conduct this experiment over a larger corpus, expanding past 5,000 research papers.

Additionally, we believe it would be interesting to construct a corpus on news data related to COVID-19 and analyze the difference between risk factors often reported in the news versus the CDC website versus the CORD-19 research data set.

7 Conclusion

We were able to show that our model learns some meaning of risk factors, as evidenced by our comparison of holdout and non-holdout tokens to our provided risk factors. Our method also finds spurious similarities with tokens unrelated to risk factors. Regardless, we believe this method shows promise for future research with additional analysis and more resources.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with lstm.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

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

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, et al. 2020. Cord-19: The covid-19 open research dataset. *arXiv preprint arXiv:2004.10706*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

A Supplemental Material

Our code used for this research is publically available on Github. The CORD-19 data set is also publically available on Kaggle. The manually defined risk factors were taken from the CDC's website.