20th Australian International Aerospace Congress

ISBN number: 978-1-925627-66-4

Please select category below:
Normal Paper ☒
Student Paper □
Young Engineer Paper □

Natural Language Processing for Identification of Ground Truth Events in Data Curation

Nathaniel C Rigoni

[1] *Data Analytics Innovations, Engineering Technologies Systems of Systems, Rotorcraft Mission Systems, Lockheed Martin*

Abstract

Identification of ground truth events surrounding the failure and removal of parts in the field involves a time-consuming process of reading through thousands of maintenance entries in order to find the correct entry containing the part and failure mode targets for research and modelling in condition based maintenance or usage based lifing. This paper identifies and explores a method of reducing the effort needed to find ground truth events in maintenance data. This data is used to identify flights relevant to the wear and use of a part being studied for modelling. The process used is a neural network that models the free and categorical language used in the maintenance entries. This model creates an n-dimensional embedding of entries which can be compared to identify similarity of the entries or to compare the similarity of entries to search terms. Creating document embeddings of maintenance entries enables users to intelligently search their data and vastly reduces the time involved in labelling ground truth. The embeddings are robust and compensate for misspellings, acronyms, synonyms, and lack of use of particular words. This method completely replaces the use of exact match text searching for data curation. Natural language processing reduces the overall cost of modelling usage based lifing and sensor events by reducing time between raw data to curated dataset in ground truth investigation.

Keywords: Machine learning, Condition Based Maintenance, Usage Based Lifing, Natural Language Processing, Part Failure, Reliability Centered Maintenance

Introduction

Developing machine learning models from sensor data, e.g. aircraft flight data, can be challenging. In cases where the activity of interest is unknown to the analyst from the perspective of the sensor readings, typically event data is curated to identify sensor data that is relevant to the problem that is being modeled. Curating this data can also be a challenge and the time it takes to hand curate can drive up the cost of model production. Reading through documentation of events to find relevant records requires time and subject matter expertise. To overcome this obstacle, we propose developing an enhanced search mechanism through the implementation of and unsupervised toolset of Natural Language Processing (NLP).

Methods

Embeddings

Embedding layers, as part of a neural network architecture, provide a massive benefit to modeling through the reduction of noise and the increased awareness of inter component relationships in the data set used for training. Embedding layers are created through the compression or expansion of encoded data into a n-dimensional space before being used and interpreted by later layers to predict values as outputs. The components for each dimension correspond to the hidden layer weights in the embedding layer. The weights of the embedding layer are driven by the loss function and the problem statement that the neural network is set up to solve. Setting up the problem statement can be a bit tricky, keep in mind that the embeddings that will be produced are those that will reduce the greatest amount of loss in the overall problem.

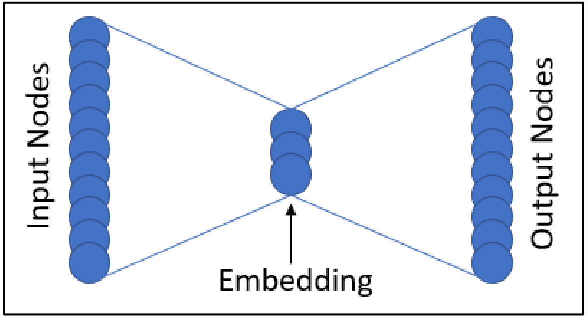

Figure 1: Encoder-Decoder Architecture

In the architecture shown in Figure 1: Encoder-Decoder Architecture, the information gets compressed before being passed to the output nodes. Information compression is useful in situations where noise must be reduced to interpret the data. Embeddings can also be used in other ways to identify similar functions in input data as it relates to the output. The compression of the information forces the embedding layer to create similar vector components for items that have similar values in the output layer. This is due to the reduced loss in predicting the same output. In curation the similarity of documents in the embedding space are compared to find documents with similar context.

Word and Document Embeddings

Utilizing an embedding layer, we can build an architecture to embed words as they are used in the documents that they occur in, gathering an n-dimensional meaning for each word based on its use. This is done using neural network architecture and a problem application known as doc2vec. Doc2vec applies a skip-gram methodology to pair a target word with a co-occurring word and a set of non-co-occurring words used as negatives. These co-occurring and non-co-occurring words are called context pairs. They are embedded in their own layer. In another layer the target word is embedded, then the 2 embedding layers are summed and try to produce the vector [1, 0, 0, 0, 0] (in this case there is 1 co-occurring word and 4 negative words). This approach embeds the context of each word into the embedding. Adding to this we also use another layer to embed the *document ID* where the target word and context pairs occur. This provides us with an embedding for the entire document.

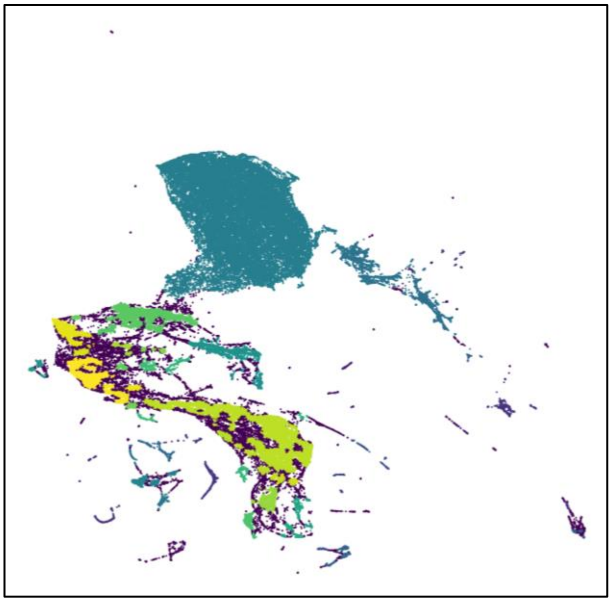

Figure 2: UMAP Reduced plot of N-dimensional Embedding Space

In Figure 2, an example of Uniform Manifold Approximation and Projection of a 500-dimensional embedding is shown. In this figure the colors represent different clusters of documents which are also known as topics. To evaluate how similar documents or words are to each other a measure of the cosine between the vectors for each is taken. With Cosine Similarity, 2 documents that have a value close to 1 are considered similar and documents whose value is close to 0 are dissimilar.

Equation 1:Cosine Similarity

$$cosine\ similarity = \frac{A * B}{\|A\|\|B\|}$$

Using cosine similarity between vector embedding, documents can now be related to each other in the n-dimensional space. This will become the basis of our search mechanism for curating data.

Topics

Utilizing the embedding space and understanding that similar documents will occupy spaces close to each other, clustering can be performed to group similar documents together into topics. These topics can then be used to segregate and curate data in a data driven and feature driven method. Figure 3 shows a breakdown of a topic space for documents from a hierarchical clustering method of the n-dimensional embeddings. The center circle is a grouping that includes all documents in the space. These groups are then broken down in a binary way based on the distance between points until there are not enough documents to break down the groups any further. At the points of the star shape are topic categories that are very specific and as we move more towards the center circle they are aggregated into more generic topics. For example, a generic topic may be “cats”, and this breaks down to more specific topics such as “house cats” and “wild cats”.

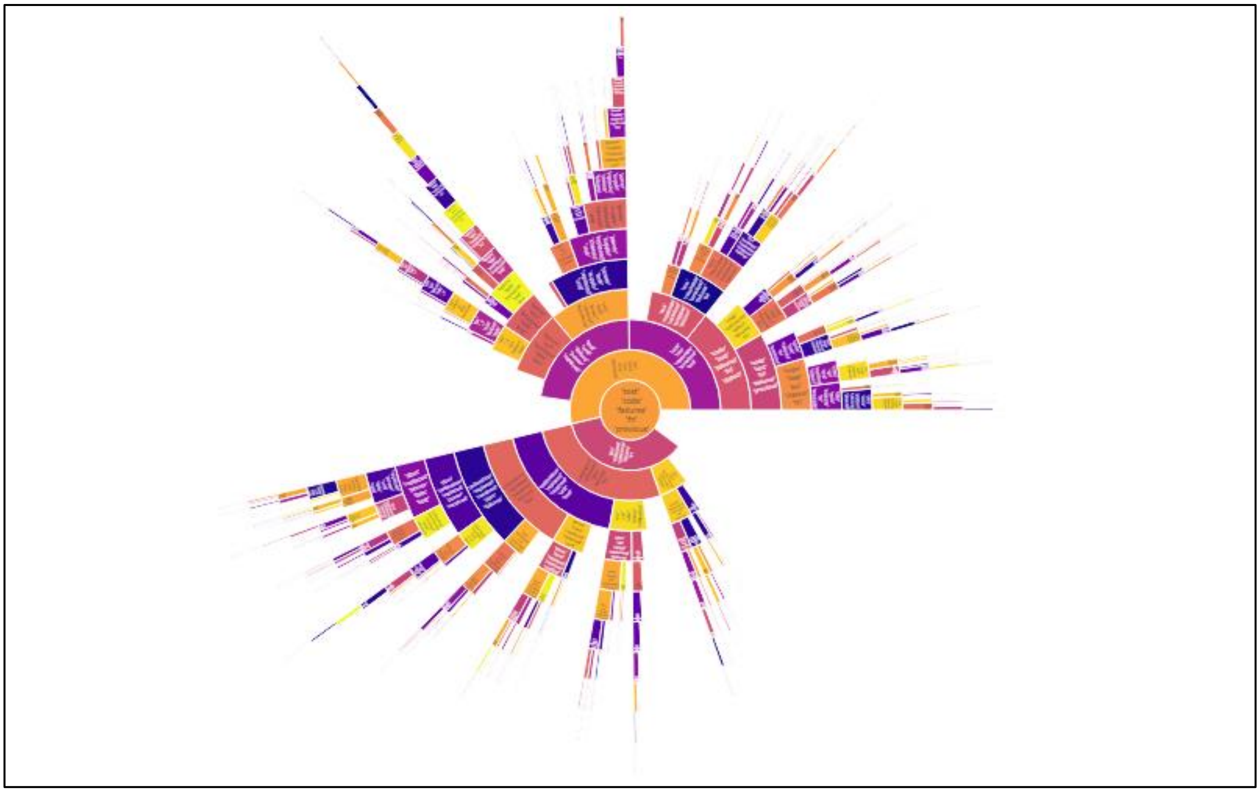

Figure 3: Hierarchical Topic Breakdown

Term Frequency Inverse Document Frequency

Term Frequency Inverse Document Frequency (TFIDF) is a method of measuring the importance of a particular term to a set of documents vs its uses in a single document. When calculating this metric it can be used to identify what is important about a single entry vs all entries or a subset of documents vs all documents. Utilizing this, search results can be analyzed for important terms that help define sub-categories present in the results.

Results

Searching Components

With the tools outline above, we construct a search workflow that reduces the number of documents in our results at each step until we have a fully curated dataset.

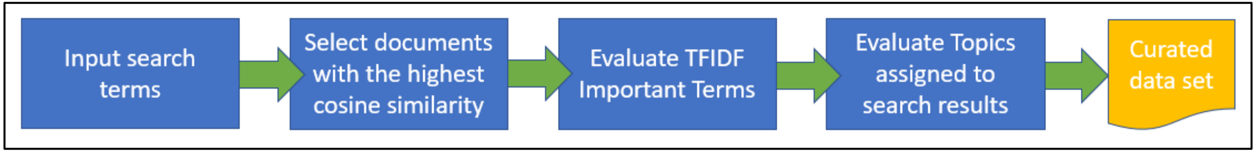
Input search terms
Select documents with the highest cosine similarity
Evaluate TFIDF Important Terms
Evaluate Topics assigned to search results
Curated data set

Figure 4: Search Workflow for Curation

Input Search Terms

To begin the search, we input our search terms and vectorize the terms using the embedding layer we trained on our documents. The search terms need to be consistent with the language used in the documents in our data set from training. Using words that do not occur in the dataset will result in words having no contribution to the context vector.

Utilize Cosine Similarity

The search vector is then used to calculate the cosine similarity of the search terms vs each document in our dataset. Sorting our results by descending similarity, we can evaluate the results to set a threshold of similarity where we will keep all documents above that number. Using this vector and cosine similarity method overcomes misspellings, colloquialisms, missing words, and even acronyms in our data search. This is because the use of these words in our data will be similar, and the model will create a similar embedding for each synonymous or related word.

Evaluate TFIDF

Calculating the TFIDF Metric on the search results, we can then evaluate if our search is covering the information that we wanted. We use the important terms to identify is there are subset cases in our results that do not apply to the curation that we intended. If so, then we use an exact match string mechanism to remove those documents from our results.

Evaluate Topics

Lastly, we evaluate the Topics that we assigned through clustering. This helps us identify groups of documents that are similar to our search criteria and may be relevant as an entire group rather than just as a subset of our results. If so, then we can append this group to our results. Once all of these steps have been completed, we can export or save the remaining documents as a curated dataset.

Conclusions

The use of this method has had a significant impact on the curation of data in model training within our organization. In one project for example it reduced the amount of time spent curating data from days to minutes and reduced the records from greater than 3 million to just over 300. In this case not all records were applicable, but it takes less time for a human to review 300 records than 3 million. Applying this method not only increases the productivity of our data science teams but releases them to do the parts of data science that they love, like training and optimizing models, rather than manually annotating text data for weeks.

References

Top2Vec, Dimo Angelov, 19 Aug 2020, arXiv:2008.09470 **[cs.CL]**