# Exercise 1: Extracting entities from unstructured data

1. Use the **Swiss Re Financial Report\*** and extract all the locations (US, Canada, Zurich, …) mentioned in the report and group them together. Create an overview on which page which location was found.
   * E.g. Canada -> Found on page 1, 2, 5, 9 ,…

**Answer** - Please find the code related to above mentioned exercise on <https://github.com/khemkaiitr/SwissRe_1/tree/master/Assignment>. The file is named as “exercise\_1\_v2.R”

1. How do you measure the **accuracy** of the solution you provide?

In order to obtain the accuracy of the solution, I will need to find the correct number of locations listed in a given pdf. I believe that without the correct number of locations (or their names), any accuracy model will not provide accurate accuracy.

1. **Generalize** the task from this particular document, to finding locations in any generic document. You can use different documents you find on the web to test your approach. Which challenges do you experience?

I tested the above code with several files (you need to change the file name) and it generalizes well. Challenges that I encountered are-

1. Some of the pdfs contained pages that were images and these could not be parsed (normally there is a layer within pdf images that contains text, so the texts are digitally typed or printed). I encountered these problems mainly in the documents from early 20th century (some old research articles).
2. Extracting location was itself not very accurate. There are various nouns and they are assigned as locations but this is a problem of the toolbox that I am using. The existing toolbox (the one I am using at least) requires significant modification.
3. Another challenge is that each document required specific preprocessing. I have tried to include several preprocessing steps to clean the document. However, there are cases when the similar names are represented in different forms and are differently spaced, which made the preprocessing difficult.

# Exercise 2. Classifying news into categories

*Assume you have a set of* ***news articles*** *from the web. You need to classify the news article into one of several categories e.g.* ***Business****,* ***Politics****,* ***Technology****,* ***Sports*** *or* ***Entertainment****.*

1. How do you build such a classifier?

Steps required to build a classifier that will classify the documents into categories-

1. Loading and reading texts – First step is to load and read all the text data. The data can be in different file format at different folders so appropriate care should be taken.
2. Preprocessing – Data is never clean so a preprocessing step is required to clean and structure the data in the appropriate format. In text preprocessing, one can remove punctuations, capitalization, numbers and common words. This process can also depend on the document type but a classifier’s accuracy can be significantly improved with the proper preprocessing steps.
3. Preparing the data for classification – Create the word dictionary from the preprocessed data by extracting the words that will be used to train the classifier. We can use word frequency of occurrence (TF) in a document as a feature vector. Another metric that is often used is TF-IDF that assigns lower weights to frequent words. In addition to the feature vector creation, add corresponding label of the document. This process is not needed if we are not interested in classifying document or texts into different categories. Now combine all the features from various document to create a feature matrix.
4. Dimensionality reduction – The feature matrix from text mining are normally very sparse so one can reduce the sparsity. Furthermore, dimensionality reduction method can be applied to reduce the features to a subset of features that contributes more to the variance in the data. Few examples are – removal of feature that has no unique value, application of PCA etc.
5. Create training and test dataset - Once the data for classifier is ready, one can train the classifier on a portion of the data and test the accuracy of the classifier on a different portion of the data. This process is important for validating the accuracy of the model and test if the model is not overfitting. We will create two segments of the data with 70% of the data as the training data and 30% as an independent test dataset.
6. Train the classifier – Once the data is prepared, the task is to choose an appropriate classifier (i.e. binary classification or multiclass classification). Few examples of binary classifiers are – two class logistic classifier, support vector machines, decision tree based models etc. For multiclass classifier one can use one-vs-all multiclass logistic regression, SVM, multiclass decision tree based model (such as random forest).

After the classifier is chosen, one can train the model and evaluate the model performance on the test data. If the performance is not optimal, one should try different algorithm, tune the hyper parameters of the classifier or look through the features.

1. What are measures to see how good it is?

One can use AUC (area under curve, mainly for binary classification), accuracy (how accurately a classifier classifies the data in to different classes), F-score (precession vs recall) and confusion matrix (best measure according to me).

1. Please prepare a full overview (including examples how you would solve this task).

Please follow the link: <https://github.com/khemkaiitr/SwissRe_1/tree/master/Assignment> . The name of the file is exercise 2.R

The model provides an accuracy of 97.91% but accuracy can be a misleading measure if the classes are not balanced. So a better measure is confusion matrix. From the script, we can observe that most of the data from the test set is classified accurately and misclassification rate is very low.

Caveats/improvements – I have not used cross-validation method. One can use cross-validation to tune the hyperparameters of the chosen model. Furthermore, if the number of samples/observations are low, one can employ bagging methods. In the end, if the model performance is not improving, I will use ensemble methods to further improve the accuracy.

# Exercise 3: General Data Science questions

Imaging you have 5 TB of PDF documents. You want to know if there any particular patterns within this documents. (E.g. Documents which are similar, documents which are duplicates, …).

Please explain the following points using no more than150 words for every point

1. How do you process the documents?

First of all, we need to read these documents and parse the words (tokenization). After the pre-processing, one can compute document term matrix (Word frequency per document) or better term frequency-inverse document frequency (TF-IDF) that measures how important a term is. One the feature vector for each document is prepared, one can compute cosine similarity index to see how similar the documents are. If the two documents are duplicates the cosine angle between these two documents will be zero while if two two documents are completely different, the cosine angle will be the largest.

Instead of cosine similarity one can also perform clustering analysis that will group similar objects in one cluster and dissimilar away from the given cluster.

1. Which visualization do you choose to illustrate the data?

Given the large amount of the data, it will be difficult to visualise all the data at once. However, I would try to use scatter plot with different size of scatter points to demonstrate the size of the cluster. Another visualise method can be hierarchical cluster method, where similar documents will be clustered together and dissimilar cluster will be separated. One can also observe frequency of different words in each document to determine the similarity.

1. Which technology do you use to perform this task. Why?

With this large amount of the dataset, I would use MapReduce. In a normal system, the data can not be loaded in to the memory (at least in R where all the data needs to be loaded). MapReduce parallelizes the process. In MapReduce, the input is broken into several smaller parts and distributes it to the several nodes (the map step). The worker nodes perform the computation and send the result back to the master node. In the reduce step, the results are combined and the output is generated.

1. How long do you assume the process will take?

Assuming a document has an average size of 10 KB. So 5 TB of the pdfs will contain roughly 50 million documents. Assuming that there are 16 cores and each core takes around 5 milliseconds to process the entire document, so the total time taken is =

0.005\*50\*10^7/16 s = 156250 s = 43.40278 h. Then there are C(n, 2) vectors to compare that will add some time. There will some extra time taken at the reducer step as well. So I assume it will take around 45 hours in total.

1. If the processing time is too long – how can you speed up the process?

Few methods to speed up the process –

* + - Increase the number of cores
    - Place the files on HDFS.
    - Smart sampling and comparison.