Text Mining/Analytics Projects

Defining a text mining/analytics task is rather difficult, primarily because problems themselves are vague (unstructured there as well). But if you think of a data mining problem (e.g., supervised and unsupervised), and apply what you would do in a (data mining) project, then you can think of what you could do in a text mining project.

Basically, for a supervised problem, you try various parameter values and multiple classification algorithms to derive an optimal classifier. For an unsupervised algorithm, you derive clusters (by hard or soft clustering) and inspect the derived clusters for meaning. You try various clusterings (number of clusters, algorithms) and find the one that derived the most meaningful results.

However, in text mining the details (what you do exactly) are very much dependent on the problem. For example, if the documents were written by users, you may consider doing spell-checking (while formal documents or newspapers won’t need that). Also POS tagging may help with sentiment/opinion analysis but generally not much for text classification.

Below I write some of my thoughts on what you could do in a text mining project (i.e., to make a project more complete/comprehensive).

1. Text Classification/categorization

This is a supervised learning.

For a text classification problem, basically you pre-process the data and feed it to a classifier and obtain the accuracy. Now about the goal(s) of the problem. Of course, the main goal of the problem is to build an accurate classifier (or ‘predictive model’). To wit, you should use several classification algorithms (at least two), and for each one you experiment with various parameters of the model. You should also experiment with various data transformation techniques, such as binary/tfidf/mutual information. You do these experiments to obtain the optimal model/classifier (where performance is compared by accuracy or any other measure, such as F1). In the report, you describe your experiments and report the results, with discussion and conclusions. That’s how classification problems are typically done and reported. Text Classification is no different.

But in addition to the standard things to do, you have more to do in text mining because the starting data is unstructured. You have to first (pre-)process the original data into a structured format. But this (pre-)processing has various ‘options’, for instance narrowing contexts (e.g., n-word window), with/without stemming/lemmatization or pos-tagging, a possible use of custom synonyms (to combine terms even more), filtering low-frequency words, adding bigrams, etc. etc. All those ‘decisions’ are dependent on the problem, and you might ‘skip’ experimenting with some of them (and go with just one setting) when it is ‘clear’ from the task or domain (or from your experience). However, those may affect the results. Therefore, you add those experiments in the problem task, and report that.

Another goal of text classification is to understand the domain of the data. For example, you can derive the important words/features – to wit, you can look at frequency, mutual information, term weights in
the model, terms used in the classification rules/trees. Also, investigation of important features leads to feature selection – to filter unimportant/irrelevant features. Then of course you can experiment with features as well. Observations and description of the experiments should be included in the report as well (unless the problem is solely to create accurate models).

So I would say a text classification task should:

- Experiment with varying text pre-processing parameters (including features).
- Experiment with various classification algorithms and model parameters.
- Inspect the resulting model (including error analysis).

And if the dataset was on the smaller side, you should be able to do a good amount of experiments and close inspection and discussion of the results.

2. Text Clustering / Topic Extraction

This is basically an unsupervised learning (i.e., mining), and typical tasks are text clustering and topic extraction (hard or soft clustering).

Generally you do text clustering to obtain groups of similar texts – to get the big picture of what the data contains. For example, you cluster user reviews and get the big picture of what the users are writing. Topics are soft-clustering as you know, and allows more ‘accurate’ picture because documents are not forced to belong to one cluster.

In clustering, you experiment with various parameter settings (e.g., number of clusters to derive, objective functions), and clustering algorithms (EM, hierarchical, Kmeans, etc.) to derive the optimal clustering. Then in the report, you describe your experiments and report the results, with discussion and conclusions.

For text mining/clustering, the same text (pre-)processing issues mentioned in the previous section applies, because that’s the step applied when the structured data was made.

A primary goal of clustering is to obtain meaningful clusters. So for that purpose, you really have to inspect the derived clusters and do a detailed analysis. Look at the actual text, but possibly other attributes of the data instances, for example the user’s gender or geo locations. And analyze the clusters made by text (only) are in some ways correlating with other attributes. Well to wit, you should have expectations too before you go in – why you expect a particular attribute to be correlated/not correlated in the domain of the data. All those things should be described in the report.

So I would say a text mining/clustering task should do:

- Experiment with varying text pre-processing parameters (including features).
- Experiment with various clustering algorithms and parameters in the algorithm.
• Inspect the resulting clusters.

Again, if the dataset was on the smaller side, you should be able to do a good amount of experiments and close inspection and discussion of the results.