Homework 2 - FAQ

Question: I'm trying to get started on homework 2, but I'm really not sure where to start. I understand that we're supposed to generate ARFF files, but I have no idea what we need to do to generate them or what attributes need to be defined in the files. Can you give any hints?

Answer: Try to extract features (integer values or strings) from the documents that distinguishes one class from others, these features should be general enough to describe the documents in the same class, not too specific otherwise your model will not generalize for the test data.

For example:

1) If you want to classify the source of the document: NWIRE vs BN, lexical cues can be used as features, e.g., the word 'tonight' may appear in the beginning of a BN show more often than in NWire, right? So this is one feature.

2) Do sentences in NWire tend to be longer than those in BN? If you think that this would be a feature add it.

3) You could try the following: take two documents randomly form our set (do not look at the tags), try to guess the source of these documents by reading them, think about the reasons that made you take the decision → then extract these features and put them in the arff file. Note that each classifier may have its own set of features.

Question: Are there any general techniques you can recommend to collect features? Is there anything specific that we were taught in class that would be helpful?

Answer: You can compute some statistics from the stories and add them as features: you can extract lexical features, syntactic features (maybe the usage of some syntactic constituents differ from one class to other), and 2-3-gram features – think about a way to represent them. There are more, just try to be creative as much as you can.

Remember: for complicated features, you have to support your intuition in the write-up, do not use bi-gram if you do not have a supporting intuition or theory.

Question: "You may extract different features for different classification tasks, but you are not required to". Even if we're not required to, are we still supposed to use different features? It doesn't seem like in most cases the same features that are needed to classify one category would be useful for another.

Answer: If you think that you found a magic set of features good enough to classify all types, just use them in all tasks. Actually, I recommend you do this first, so you will be done with the homework, and later on if you have time try to explore complicated and more task dependent features for the tasks that you did not do well with the initial set of features.

Question: "For some of the classifications every story in a document will have the same class". Therefore, it might make sense for every story from a document to have an identical feature vector. I'm not sure why this would be the case. Even if they are the same class, they're still likely to have different features. Why would it help to use the same features for all of them?

Answer: **NO,** you will have a feature vector for each story, because you want the feature vector to represent the content of one story, not the content of other one. Hopefully, the values of *some* of the features will be identical (or "similar" values if they are numbers) across the stores from the same class, and some features will have different values. Otherwise, it may show that your choice of features was not good.

Question: Do you recommend using any other APIs besides weka for this assignment? Something that might help with feature extraction?

Answer: Weka is a set of machine learning (ML) algorithms, you will use weka only to train your classification models and run the cross-validation experiments -- not to extract any feature. You can use any library you like (obviously, except a document classifier or another ML toolkit). Your program should extract the features and produce the arff files...(then use them in weka).

Question: Quick question about the 10 fold cross validation. Does this mean that we should just use the ten fold cross validation option during classification/training to build the model using the training data such that we should use the model that displays more accuracy?

Answer: Yes exactly, this is what you should do.

Question: You said we can use the output of a previous classification to help with another classification. It seems that to accomplish this, I would need to input a single feature vector to weka and have weka return a classification type. Is this what would I need to do? If so, can you give any hints on how to do this?

Answer: What you said is correct! To do that, for example let's assume that we want to use the source type to classify language given story A.

1) Extract one feature vector from story A (using the feature extractor for source type)
2) build an arrf file that includes this vector.
3) Run weka using the command line giving it the source type model and this arff file => weka will output the source type.
4) Run your feature extractor for the language given this source type and the story A, it should build your new feature vector for story A. write it in your arff file.

Please refer to Weka command line help for the syntax.

Question Do you recommend any specific classification models? NaiveBayes was demoed in class, but other than that I have no idea how to differentiate any of them (other than trying each one manually and see which one gives the best results).

Answer: This is a good question**;** the answer to this question is in the Machine Learning course. NaiveBayes has its limitation because of the independent assumption; therefore it may not do well in this task, because your features may not be independent random variables. Try whatever available in weka. Try J48 tree, JRip (rule based approach), SVM (SMO: this is a very strong classification algorithm, it's based on structural risk minimization – it should do well on unseen data, by principle) even though it might take a lot of time to train your model. Multilayer preceptor – using empirical risk minimization… In this class, you're required to experiment these algorithms but not to deeply understand them – you should read a bit about your classifier to support your choice in the write-up.
