**How to add user-defined features**

**Description**

1. Add a new column to the train data with a new feature.
2. Update the prop file with ‘map= word=0, answer=1, mySpecialFeature=2’.
3. Add a new annotation.
4. Add its type to this table in ‘edu.stanford.nlp.ling.AnnotationLookup’.
5. If you modify AnnotationLookup, you need to read the data from the column, translate it to the desired object type, and attach it to the CoreLabel using a CoreAnnotation. Quite a few CoreAnnotations are provided in the class appropriately called CoreAnnotations. If the particular one you are looking for is not present, you can add a new subclass by using one of the existing CoreAnnotations as an example.
6. If the feature you attached to the CoreLabel is not already used as a feature in NERFeatureFactory, you will need to add code that extracts the feature from the CoreLabel and adds it to the feature set.
7. features must have unique names, or they will conflict with existing features, which is why we add markers such as "-GENIA", "-PGENIA", and "-NGENIA" to our features.
8. As long as you choose a unique marker, the feature itself can be any string followed by its marker and will not conflict with any existing features.

**Demo**

1. Modify the properity file:
   1. map = word=0,answer=1,myUDFeature=2 (myUDFeature is the feature name I added and the program will find the feature use this word as key.)
   2. useUDFeature=true (useUDFeature is the flag defined in SeqClassifierFlags which can map the usage trigger to annotation)
2. Add a flag in SeqClassifierFlags.java
   1. public boolean useUDFeature = true; (Add at the variable declaration area in the class)
   2. **else** **if** (key.equalsIgnoreCase("useUDFeature")){

//UDF-Panda

useUDFeature= Boolean.*parseBoolean*(val);

}

Add this code in the function - public void setProperties(Properties props, boolean printProps)

1. Add a new class at CoreAnnotations.java
   1. public static class myUDFeature implements CoreAnnotation<String> {

public Class<String> getType() {

return String.class;

}

}

* 1. …

1. Add a Key LookUp method at AnnotationLookup.java
   1. MY\_UDF\_TAG(CoreAnnotations.myUDFeature.class,"myUDFeature") (Add in private enum KeyLookup{})
2. Add and use the udf feature in NERFeatureFactory.java
   1. **if**(flags.useUDFeature){

featuresC.add(c.get(CoreAnnotations.myUDFeature.**class**)+"-my\_udf\_tag");

} ( this code add in the function protected Collection<String> featuresC(PaddedList<IN> cInfo, int loc), we can change the function as featuresCpC() if needed)

1. Run the CRFClassifier.java by the way of ‘Run as configuration’ and test the new model trained with udf features.