6.864 Final Project: Recipe Generation from Experimental Procedures

We approach Recipe Generation from Raw Text as a multi-step problem.
Step 1: Tag words using our tagging scheme
POS Tagging:
N - not tagged as anything special
A - Action performed (generally past tense e.g. were dissolved) in a step
I - Ingredient used in a step
E - Equipment used in a step
PR - Product of a step
P - Property, something that describes an Ingredient, Equipment, or Action
R - Replaces an Ingredient or Equipment in a step with a reference to the previous step (e.g. uses the previous product)

Example:

In_N brief_N ,_N a_I 2:1_I mixture_I of_N Pluronic_P surfactant_P F127_P and_N phloroglucinol_P (3.78_P g)_P and_N hydrochloric_P acid_P 37_P wt_P %_P aqueous_P solution_P (0.5_P g)_P were_A dissolved_A in_N 65_I mL_I of_N absolute_P ethanol_P.

The_R solution_R was_A then_N heated_A to_N 80_P C_P with_N vigorous_N stirring_N.

Step 2: Group words that are tagged with the same tags and remove words tagged with N. Words that are close by but separated by one word tagged by N (e.g. and_N or then_N) are combined with a (+) indicator. We also separate out Action groups such that each sentence contains at most 1 action group. We also remove all sentences without any actions.

Example:

(The solution, R) (was (+) heated, A) (80 C, P)

Step 3: Generate Dependencies. We want to see what each group of words modifies so that we can have an order to our steps. We use a transition based dependency parser that uses the shift/left/right arc technique to generate our dependencies. In general, modifiers (I, E, PR, R, P) will depend on the Action. Each modifier may again be modified by a Property P.

Step 4: Generate Recipe from dependencies. Using the dependencies we generate a json-like object that encapsulates the relationship between the modifiers and action.

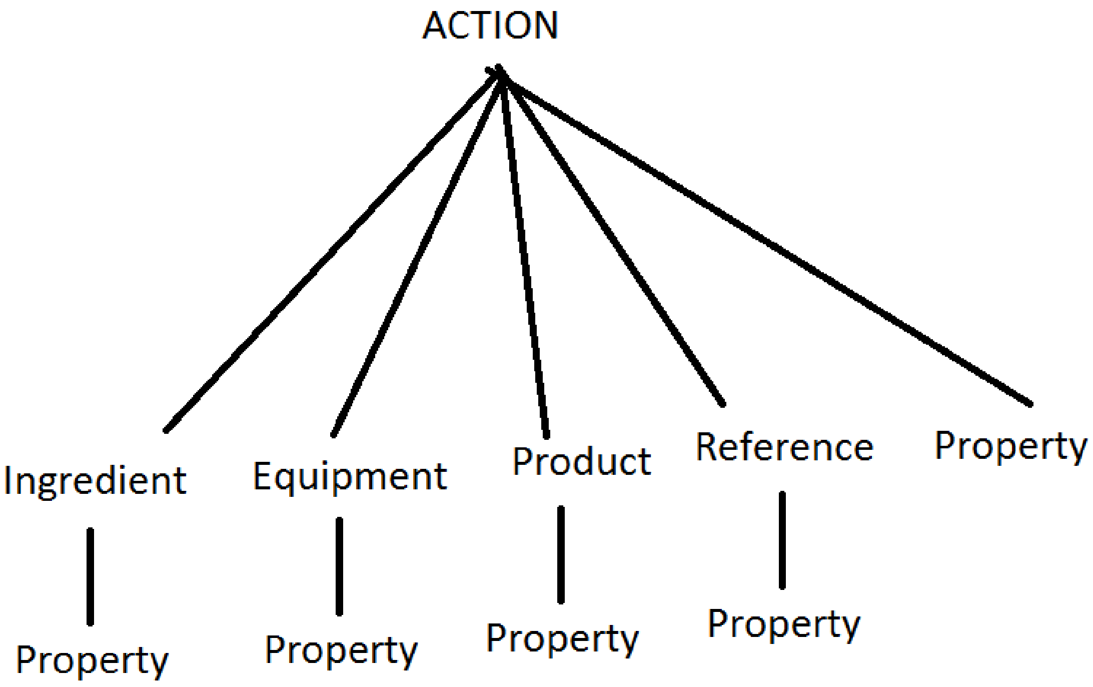
Recipe Format:
Each sentence will look like dict.

```
{"A": [("were ground", {"PR": [("electrodes", {})], "I": [("Li2S particles (+) Super P carbon black",
{}), ("polyvinylidene fluoride", {})], "E": [("mortar", {"P": ["10 min"]})], "P": [("weight ratio of
40:45:15", {})]})]}
```

The action is at the top. It contains a list of tuples of the form (“action string”, dict of modifiers)

The dict of modifiers contain each possible modifier as shown in the recipe diagram below. Each modifier is again a list of tuples of the form (“modifier string”, dict of property modifiers). The dict of property modifiers is a list of properties that modify the modifier.

RECIPE DIAGRAM:


ACTION
Ingredient
Equipment
Product
Reference
Property
Property
Property
Property
Property