# L4 Project Progress Report

Student: Lovisa Sundin 2092502 • Supervisor: Mary Ellen Foster

## Project Description

This is a self-defined project aimed at creating a parser and diagram-generator for cooking recipes. The idea is that a natural-language, semi-structured recipe that conforms to some simple formatting rules can be pasted into a textbox in a web application interface. Under the hood, key action verbs and dependencies would be extracted, lemmatised into standard dictionary form and matched to a manually constructed ontology of keywords that have accompanying pictograms.

The recipe will be parsed and visualised on a sentence-by-sentence basis. It will *not* be represented holistically as a graph of dependencies that would allow us to, for example, trace how ingredients are combined and transformed over time, or perceive the parallelisms often inherent in a cooking sequence. Analysis of recipes has suggested that sentences constitute a meaningful communicative unit in recipes, typically corresponding to a mentally self-contained action.

Once pictograms have been identified for a particular sentence, information about within-sentence dependency relationships will be used to combine the pictograms in an informative diagram. For example, “add flour, milk and eggs” would have an “add”-symbol juxtaposed with images for the three ingredients. Cardinal numbers (e.g. temperatures, quantities) will also be made visually salient, and to prevent loss of information the diagram will be annotated with the original sentences. The sentence-diagrams will then be presented linearly, forming a “visual recipe” that the end-user can print out.

Initially the idea was parse all kinds of recipes, but in order to reduce the variability of actions and ingredients (thereby increasing the likelihood that there is an available diagram or match in a manually constructed keyword ontology), it was decided that the parser would be confined to baking recipes. By expanding the ontology and diagram database, it should in practice be easy to extend to other types of cooking, as there are no obvious grammatical differences to take into account.

The source of recipes will be Epicurious.com, which is one of the biggest online recipe services. Its baking recipes are relatively complex, often containing obscure ingredients beyond the pantry stables, however they have the virtue of being structurally rather uniform with sentences starting almost always with an imperative verb. Exotic ingredients can often be mapped to one of the more ambiguous diagrams.

## Progress

The first step was to review the literature and existing efforts in parsing and visualising recipes. A major task was deciding the generality of the parser and what structural assumptions it would do about the input text. If the input were too formatted, like the range of recipe markup-languages available, then the task would become trivial and the application useless, as existing recipe banks would have to be manually re-structured. Yet, it would be intractable to provide for any contingency or typographical error. For linguistic analysis, a set of 20 baking recipes was therefore

chosen quasi-randomly from Epicurious (with a preference for recipes that didn't incorporate too many obscure ingredients) and minor errors and deviations in format were manually corrected.

Next, in order to know which information would be necessary to extract, a prototype was done for how the ideal diagrammatic output would look like for a specific recipe. This clarified parsing objectives and the required data structures.

I have learned Illustrator in order to do the vector graphics for the diagram, and have kick-started the production. My pictogram-illustrating rate so far indicates that it is feasible to produce the required number of diagrams in time and that it is possible, albeit challenging, to make informative pictograms for abstract verbs like "combine" or other actions that do not readily lend themselves to visualisation. Pictogram-illustration occurs in parallel with other project activities.

The next step was to evaluate and familiarise myself with the two major natural language processing toolkits on the market: the Stanford CoreNLP and NLTK. What I have found is that NLTK is more accurate in part-of-speech tagging of ingredient-listings, but that Stanford – which has been trained on corpora containing more imperatives – outperforms NLTK in instruction-parsing. I have also studied a range of NLP and grammatical concepts to understand the various relations.

The bulk of the parsing functionality has been implemented, combining the two toolkits. Accuracy can be perfected for individual recipes, but performance across the entire sample set is not yet satisfactory and needs polishing. Due to the hurdle of identifying the key verb in a sentence, I have not yet implemented the final "binding" step where all of the information is collated in a single data structure for diagram generation. I have not yet begun to implement the diagram generator or decide about the precise geometric relationships, but have decided that it will be done in HTML5 Canvas.

## Plan

The first step is to fine-tune and enhance the accuracy of the implemented parsing functionality. I will have to systemise this procedure and run it on all of the sample recipes, and finalise the data structure format. I will also have to encapsulate the source code and make sure it adheres to software-engineering principles like coupling and cohesion, as well to review its efficiency. My goal is to have this finished by Christmas.

The other major step is to learn HTML5 Canvas and calculate how to arrange the pictograms on the plane. I think I will need the first half of January to complete this functionality and the remainder of the diagrams. If time allows, I could deploy this as an online web application.

The second half of January would be dedicated to running user evaluations. Evaluations are planned to proceed as follows: a "test pool" of recipes is chosen from Epicurious (again with imperfect randomness), manually corrected and diagrams generated. The raw recipe (or markup) is never made visible to the user: instead, he or she will be asked to reverse-engineer the sentence that a diagram represents by writing it down, so that I (or another user) can rate the similarity to the raw recipe sentence.

The whole of February needs to be dedicated to dissertation write-up.

## Problem

The accuracy of my parser will ultimately be bounded by the capabilities of these two processing toolkits, as the project will not be concerned with machine learning. The toolkits have not been trained on recipes, and this is very evident in their outputs. I have had to resort to a number of guesswork-based hacks and heuristics, such as prefixing imperatives with a “We” to prevent them from being tagged as nouns.

As I see it, the most challenging aspect is deciding which verbs and nouns are most informative and which verbs would only clutter the diagram by being made salient. Often, a sentence would require several diagrams, it is challenging to programmatically comprehend if a verb is essentially informative of action, or a descriptor that can be discarded.

There may be potential difficulties in implementing the diagram generator even when final data format has been clarified, however they are difficult to anticipate at the moment and this is expected to be the easier step.

Currently my implementation, when run on my laptop, is prohibitively slow, with a single recipe taking 2-3 minutes. Most of this duration is unavoidable due to the computation-heavy nature of NLP, however I will have to consider how to organise my modules and algorithms as efficiently as possible and exploit any opportunities for parallelisation so that the problem could be alleviated by the availability of more processing power.