Text mining steps

Step one	Self-defined goals will be formatted as *comma-separated values*. This means, first self-defined goals will be parsed into their separate components using tokenizer algorithms as part of *nltk* library. For example, if a person writes "Walk outside 3-4 x a week in the morning for 15-20 min", the sentence will be separated into tokens (words) as below: ['Walk', 'outside', '3-4', 'x', 'week', 'in', 'the', 'morning', 'for', '15-20', 'min'] Using *pandas* library, this information will be transferred into an in-memory data structures.
Step two	Having the goals separated into tokens will give the opportunity to: (i) identify the role of each word in the sentence; and (ii) evaluate if the word is part of the goal setting lexicon. Parts of speech (POS) tagging functions such as *pos_tag()* will be used to tag each token into their corresponding role (class). Each class has an associated code. To illustrate how the results of this step will look, the *pos_tag()* algorithm was applied to the goal example above: [('Walk', 'VB'), ('outside', 'IN'), ('3-4', 'JJ'), ('x', 'JJ'), ('week', 'NN'), ('in', 'IN'), ('the', 'DT'), ('morning', 'NN'), ('for', 'IN'), ('15-20', 'JJ'), ('min', 'NN')] However, the POS tagger does not always give the best results and may contain inaccurate results. To deal with this inaccuracy, different tagging algorithms available from nltk library[1] will be tried and the one with the highest accuracy will be selected.
Step three	At this stage, all the tokens in persons' goals have been tagged. The next step is to evaluate tokens based on the previously gathered lexicon. Regular expression algorithms will be used to match tokens to the initial lexicon. If a match was found, that word is classified as a "matched" word. For example, in the case of our example above: line = Walk outside 3-4 x a week in the morning for 15-20 min search = re.search(r'walk', line, re.M|re.I) if search: print "Found!"

[1] https://www.nltk.org/api/nltk.tag.html