README for Arguing Lexicon (2007)

The lexicon described here was hand built as described in the paper. Please refer to the paper for more details on the construction, intuition etc.

The distribution contains 22 *.tff files.

17 of theses list patterns that represent arguing. Each file represents a type (category) of arguing discussed in the paper. For example, "assessments.tff" lists linguistic patterns that are commonly found when someone uses assessment to argue for something. Similarly, "emphasis.tff" lists linguistic patterns that indicate arguing via drawing emphasis towards the idea being argued for. Examples of entries from "emphasis.tff" are "clearly", "of course" and "without a doubt"

Here are the list of files containing arguing patterns

1. assessments.tff
2. doubt.tff
3. authority.tff
4. emphasis.tff
5. necessity.tff
6. causation.tff
7. generalization.tff
8. structure.tff
9. conditionals.tff
10. inconsistency.tff
11. possibility.tff
12. wants.tff
13. contrast.tff
14. priority.tff
15. difficulty.tff
16. inyourshoes.tff
17. rhetoricalquestion.tff

Note that all lexicon entries are in the form of regular expression patterns. While many entries are simply n-grams (e.g. "clearly", "without a doubt"), there are others which employ regular expressions and macros. (We use "#" to denote comments in all the files)

Some standard regular expression employed are:

?: Question mark is used to indicate matching 0 or 1 time.

(): parentheses are used to group (or bind) multiple words together. This is used to indicate that bi-grams or any higher order ngrams should be treated as a single unit.

| : **alternation** is used to mean that either the left hand OR right values can be used, for example, the following

i want to (highlight|emphasize|underscore) should match all of the following :

“i want to highlight”

“i want to emphasize”

“i want to underscore”

and “what is ((gonna)|(going to)) happen is”should match

what is going to happen is

what is gonna happen is

In general, we use the standard conventions for regular expressions (e.g. http://en.wikipedia.org/wiki/Regular_expression)

The linguistic patterns for lexicon entries also use macros. These are words that start with a “@”. Marcos expand to a set of words, and this expansion is provided in a separate macro file.

There are 5 macro files:

1. modals.tff
2. spoken.tff
3. wordclasses.tff
4. pronoun.tff
5. intensifiers.tff

For example, the following entry in the modals.tff file gives all the words that should, one by one, replace every occurrence of @BE in all lexicon patterns.
@BE={be,is,am,are,were,was,been,being}
Hence the entry “(@BE) bound to” in emphasis.tff expands to
be bound to
is bound to
am bound to
are bound to

were bound to
was bound to
been bound to
being bound to

Similarly, the pattern “(@BE) able to” from possibility.tff expands to

be able to
is able to
am able to
are able to
were able to
was able to
been able to
being able to

The macros also sometimes specify the part of speech. For example, the following two macros differ only because one specifies matching with verbs and the other with nouns.

1. @EMO2V={hate, dislike, disprefer}
2. @EMO2N={hate, dislike, dispreference}

Macros were used so that we could expand the lists independently and separately of the actual patterns in the lexicons. Thus, every time a new word was added to a macro’s list (e.g. “being” was added to @BE), we did not need to change multiple lexicon patterns exhaustively.

TIP: When reading the lexicons, it might make sense to read the files with the macros first, and store the entries are key-value pairs in a hashmap (e.g. @BE would be a key and its expansions would be be, is, am, are, were, was, been, being). Then, while reading the lexicon patterns, the expansions and substitutions for the macros can be performed on the fly. The resulting expanded lexicons can be used for pattern matching against your text corpus.