Some natural language processing terminology

M. Covington 2008 Aug. 29

Parts of a natural language processing software system (not all systems have all of these):

Tokenizer – Component that breaks a text up into words,
e.g., “dog and cat” --> {“dog”, “and”, “cat”}

Issues:
(1) How to treat contractions, e.g., “don’t” --> “do” “n’t”
(2) Correct handling of strings such as “$123,456.78”
(3) Distinguishing end-of-sentence period from abbreviation period
(this is probably impossible in general, at this level)
(4) Whether to convert lowercase to uppercase
(tagging goes better if you don’t)

Stemmer – Component that removes suffixes (and, in some languages, prefixes)
e.g. “dogs chased cats” --> “dog” “chase” “cat”.

The goal of stemming is not perfect accuracy – it is to make related words look alike. The output of a stemmer may be something that is not a correctly spelled word, but serves the purpose.

Issues:
(1) How much to remove. The Porter Stemming Algorithm is very heavy-handed. For many purposes, however, we only want to remove the most productive suffixes: -s, -ed, -ing.
(2) When to do it, if at all. Stemming interferes with tagging.

Tagger – Component that labels words as parts of speech (noun, verb, etc.) without doing a full analysis of sentence structure, e.g.
“dog and cat” --> “dog/NN and/CC cat/NN”

Issues:
(1) What system of tagging to use. Most people use the Penn Treebank.
(2) How much accuracy is possible. 95% typical.
(3) Whether errors matter. E.g., the difference between verb past tense and verb past participle is arguably a matter of syntax, not tagging.
(4) What method of tagging to use. I usually use a homegrown adaptation of that published by Brill in *Computational Linguistics* 1995.

Lemmatizer – Component that reduces each word to its dictionary form (lemma), e.g.,
“dogs chase cats” --> “dog” “chase” “cat”.

Unlike a stemmer, a lemmatizer strives to produce *correct* results and
Relies heavily on a dictionary.

Lexicon – A dictionary; a database of words and (some of) their attributes.

Morphological analyzer – Component that interprets word forms, e.g., (in Spanish) “secuestraron” –> secuestrar, verb, 3rd person, plural, preterit tense, indicative mood. For English this is often combined with tagging since there isn’t much to it.

Issue: Where to draw the line between inflectional morphology (forms of existing words) and derivational morphology (formation of new words using suffixation and similar processes), and whether to tackle the latter.

Parser – Component that recovers the sentence structure, e.g., “All dogs chase cats” --> noun phrase “all dogs,” verb phrase “chase cats” consisting of verb “chase” and noun phrase “cats.”

This is a technically challenging problem, but shallow (incomplete) parsing can be fast, accurate, and useful.

Semantic interpreter – Component that recovers a representation of the meaning of the sentence. This is a very challenging problem, but there are ways of doing it incompletely that are useful. It is of course much more than just the meanings of the words.

Other aspects of NLP

Information retrieval – The task of finding texts (in a collection) that are about a particular subject or are about the same subject as a given text. Comparison of relative word frequencies is the usual approach.

tf*idf – Term frequency times (logarithmic) inverse document frequency.
A way of weighting vocabulary items for comparison so that words that occur in large numbers of texts are given less weight.

Latent semantic indexing – An indirect method of comparing texts for information retrieval. Besides the texts to be compared, it uses a large knowledge base indicating what words are likely to occur in the same texts. Thus, a text containing word A but not word B might be rated as similar to a text that contains B but not A, if the knowledge base knows that nearly all the *other* texts that contain either A or B contain both of them. This is implemented with matrix arithmetic and the knowledge base is a matrix.

Text categorization (document classification) – The task of sorting texts by subject matter or other attributes.