Word Sketches

Word sketches are one-page automatic, corpus-derived summary of a word's grammatical and collocational behaviour. A simplified word sketch for the English noun *flour* is:

The Sketch Engine is a corpus query system which allows the user to view word sketches, thesaurally similar words, and 'sketch differences', as well as the more familiar CQS functions. The word sketches are fully integrated with the concordancing: by clicking on a collocate of interest in the word sketch, the user is taken to a concordance of the corpus evidence giving rise to that collocate in that grammatical relation. If the user clicks on the word toast in the list of high-salience objects in the sketch for the verb spread, they will be taken to a concordance of contexts where toast (n) occurs as object of spread (v).

flour *noun*

- OBJECT_OF sift sieve grind mix add raise produce put
- ADJ_MODIFIER self-raising wholemeal seasoned plain white organic fine strong
- NOUN_MODIFIER wheat soya tbsp maize corn rice bread cup
- MODIFIES tortilla milling mill mixture
- AND/OR salt butter sugar flour cook rice bread cereal egg wheat grain powder
- PP_INTO bowl

Flour *noun*

ADJ	**strong | plain, self-raising white, wholemeal | stone-ground | unbleached | rice, rye, wheat, etc.**
QUANT	**bag, packet, sack**
VERB + FLOUR	**use | add, blend, fold in, mix (in), rub sth in/into, stir (in) | sieve, sift**
FLOUR + NOUN	**mill**

Grammatical relations

In order to identify the grammatical relations between words, the sketch engine needs to know how to find words connected by a grammatical relation in the language in question. The sketch engine countenances two possibilities.

In the first, the input corpus has been parsed and the information about which word-instances stand in which grammatical relations with which other word-instances is embedded in the corpus. Currently, dependency-based syntactically annotated corpora are fully supported. Phrase-structured trees need heads of phrases to be marked.

In the second, the input corpus is loaded into the sketch engine unparsed, and the sketch engine supports the process of identifying grammatical relation instances. In this approach, we distinguish two roles: a regular user such as a lexicographer, and an expert user, ideally a linguist with some experience and familiarity with computational formalisms. The expert user will then define each grammatical relation, using the sketch engine to test and develop it, and will load the grammatical relation set into the sketch engine. The sketch engine will then find all the grammatical relation instances and give all users access to word sketches.

The formalism for the grammatical relations is the formalism used for all searches that a user (expert or regular) might make on the corpus. It uses regular expression over POS-tags. An example: if we wish to define the English verb-object relation, we first note that, lexicographically, the noun we wish to capture is the head of the object noun phrase, and that this is generally the last noun of a sequence that may include determiners (DET), numbers (NUM), adjectives (ADJ) and other nouns (N). We also note that the object noun phrase is, by default, directly after the verb in active sentences, and that the lexical verb (V) is generally the last verb of the verb group. Adverbs (ADV) may intervene between verb and object. Taken together, these give a first pass definition for a “verb-object” pair, as “a verb and the last noun in any intervening sequence of adverbs, determiners, numbers, adjectives and nouns”. In the Sketch Engine formalism, using the tags given in brackets above, this is

1:”V” “(DET|NUM|ADJ|ADV|N)”* 2:”N”

The 1: and 2: mark the words to be extracted as the first and second arguments of the grammatical relation. |, (), and * are standard regular expression metacharacters. | is for disjunction and * indicates that the preceding term (here, the bracketed disjunction) occurs zero or more times.

The expert defines each grammatical relation in this way. Clearly, they need to be conversant with both the tagset and the grammar of the language. As the grammatical relations query language is the standard one for the CQS, they can use the CQS to test grammatical relation definitions and the process of grammatical relation development is well-

supported. A definition can have multiple clauses: in our work on English, we have used separate clauses for objects realized as subjects of passives, and nouns which are objects of a verb in a relative clause. Czech sketches define several clauses to capture verbal modifiers in different grammatical cases.

While there are no limits to the sophistication with which one might define a grammatical relation, we have found that very simple definitions, such as the one above, while linguistically unsatisfactory, produce very useful results. While a simple definition will miss grammatically complex instances, it is generally the case that a small number of simple patterns cover a high proportion of instances, so the majority of high salience collocates are readily found, given a large enough corpus. Our use of word sketches to date suggests that POS-tagging errors are more frequently the source of anomalous output than weaknesses in the grammar. The use of sorting based on salience statistics means that occasional mis-analyses rarely result in wrong words appearing in collocate lists.

Verb-object, while frequently the most significant grammatical relation for describing the behaviour of nouns and verbs, is also a relatively complex one to identify. Others such as the relation between an adjective and the noun it modifies (which is usually the most significant one for adjectives) or between a word and others of the same word class that it occurs in conjunction with (fish/chip; hope/pray; big/fat), or between a content word and a following preposition, are generally simpler.

These kinds of methods have been widely used; a series of workshops on Finite State methods have been among the places at which Finite State (including regular-expression) approaches to grammatical analysis have been studied. Researchers such as Gahl (1998) have explored sophisticated syntactic querying within a CQS using the same formalism.