

This quick reference gives a concise overview of the most commonly needed features of Simple Query Syntax; see Chapter 6 of Hoffmann *et al.* (2008) for a comprehensive reference and tutorial. Query expressions that you can enter in *BNCweb*'s search box are printed in typewriter font, followed by an arrow and the matching words or word sequences in italics (e.g. `st?ing` → *sting, stung*).

Basic word form searches

- To search for word forms, simply type them into the query field and click [Start query]: `glitterati` → *glitterati*
- Use wildcards for unspecified letters, and prefix or suffix searches:

? for a single arbitrary character
`s?ng` → *sing, sang, song, ...*

* for zero or more characters
`*able` → *able, table, capable, suitable, available, ...*

+ for one or more characters
`+able` → *table, capable, suitable, ...* but not *able*

??+ for three or more characters, etc.
`??+able` → *capable, ...* but not *able, table, unable, stable*

- Combine multiple wildcards: `*oo+oo*` → *Voodoo, schoolroom, ...*
- Protect wildcards and other metacharacters with backslash `\` to match the literal character (called "escaping" the metacharacter):

`\?` → *?*
`?` → *a, b, c, ..., A, B, C, ..., 1, 2, 3, ..., ., !, ?, ...*

Simple Query Syntax uses the following metacharacters:

`? * + , : @ / () [] { } _ - < >`

- List comma-separated alternatives (optionally including wildcards) in square brackets:

`??+[able, ability]` → *capable, capability, availability, ...*
`neighbo[u,]r` → *neighbour, neighbor*

- Searches are case-insensitive by default: the queries `bath`, `Bath` and `BATH` find the same matches (viz. the three word forms *bath*, *Bath* and *BATH*). Set the "Query mode" drop-down menu to "Simple query (case-sensitive)" to distinguish between *AIDS* and *aids*, for example.
- Use `:d` modifier to ignore accents: `fiancee:d` → *fiancée, fiancée* (for details, see Hoffmann *et al.* 2008, Section 6.10 and Appendix 4).

Matching parts-of-speech (POS)

- Search for a word form with a specific POS tag by linking them with an underscore `_`. Wildcards can be used both for word form and POS tag:

<code>lights_NN2</code>	➔ plural noun <i>lights</i> , but not the verb form <i>lights</i>
<code>*ly_AJ0</code>	➔ adjectives ending in <i>-ly</i> (e.g. <i>daily</i>)
<code>super+_V*</code>	➔ verb forms starting with <i>super-</i>

- You can also search by POS tag only: `_PNX` ➔ any reflexive pronoun
- Complete listing of POS tags used in the BNC is given on last page.
- Use simplified POS tags enclosed in curly braces: `super+_{VERB}` for verb forms starting with *super-* (no wildcards allowed in simplified tags).
- List of simplified POS tags (Table 3.8 of Hoffmann *et al.* (2008) shows comparison with full tagset):

<code>A, ADJ</code>	adjective	<code>INT, INTERJ</code>	interjection
<code>N, SUBST</code>	noun	<code>PREP</code>	preposition
<code>V, VERB</code>	verb	<code>PRON</code>	pronoun
<code>ADV</code>	adverb	<code>\$, STOP</code>	punctuation
<code>ART</code>	article	<code>UNC</code>	other / uncertain
<code>CONJ</code>	conjunction		

- Keep in mind that part-of-speech tags have been assigned by an automatic software tool and are not always correct (try e.g. `beer_{N}` `can_{N}`).

Headword and lemma queries

- Search by headword, enclosed in curly braces: `{light}` finds the forms *light*, *lights*, *lit*, *lighted*, *lighting*, *lighter* and *lightest* (but not the nouns *lighting* and *lighter*).
- In *BNCweb*, the lemma is a combination of headword and simplified POS tag, separated by a slash `/`. A lemma query distinguishes e.g. between the noun, verb and adjective reading of LIGHT:

<code>{light/V}</code>	➔ <i>light</i> , <i>lights</i> , <i>lit</i> , <i>lighted</i> , <i>lighting</i> (tagged as verb)
<code>{light/N}</code>	➔ <i>light</i> , <i>lights</i> (tagged as noun)
<code>{light/A}</code>	➔ <i>light</i> , <i>lighter</i> , <i>lightest</i> (tagged as adjective)

Word sequences

- Queries can consist of multiple words, e.g. `talk of the town`
- All words and punctuation symbols ("tokens") are separated by blanks; possessives (*Peter's*) and contracted forms (*they've*, *gonna*) must be split:

`he will \, wo n't he \?` → *he will, won't he?*

- Each query item in a sequence can make full use of wildcards, part-of-speech constraints, and headword or lemma searches:

`{number/N} of _{A} _NN2` → *numbers of younger men, ...*

- Use `+` to skip an arbitrary token, or `*` for an optional token. Combine `+` and `*` for larger gaps, e.g. `++++*` to skip between 3 and 5 tokens.

`{eat} * up` → *eat up, ate up, eat it up, eaten all up, ...*

`{eat} + up` → *eat it up, eaten all up, ...* but not *eat up, ate up*

`{eat} ++* up` → *up* at a distance of 3 or 4 tokens after *eat*

Advanced lexico-grammatical patterns

- Use regular expression notation (Hoffmann *et al.* 2008, Sections 6.8 and 12.4) for alternatives, optional elements and repetition within a sequence:

<code>(_{A})?</code>	optional adjective
<code>(_{A})*</code>	zero or more adjectives (optional)
<code>(_{A})+</code>	one or more adjectives (non-optional)
<code>(_{A}){2,4}</code>	between two and four adjectives
<code>(...)</code>	matches one of the alternatives indicated by ...
<code>(...)*</code>	alternatives with repetition (optional)
<code>(...)+</code>	alternatives with repetition (non-optional)
<code>(...){2,4}</code>	between two and four repetitions of the given alternatives (may be mixed in any order)

- Regular expression notation can be nested to match complex patterns:

`the (most _AJ0 | _AJS) {man}`

→ *the biggest men, the most attractive man, ...*

`the (most (_AV0)? _AJ0 | (_AV0)? _AJS) {man}`

→ plus: *the very richest men, the most supremely stupid men, ...*

- Complex syntactic patterns can be formed, e.g. for a prepositional phrase:

`_ {PREP} (_ {ART}) ? ((_ {ADV}) ? _ {A}) * _ {N}`

"a preposition; followed by an optional article; followed by any number of adjectives (zero or more), each of which may optionally be preceded by an adverb; followed by a noun"

XML tags

- XML start and end tags can be inserted in query expression to match the boundaries of a region, e.g. the start (<s>) or end (</s>) of an s-unit:

<s> but → s-unit beginning with *but* (or *But*)
_{ART} </s> → article at end of s-unit (mostly errors)

- To match a complete region, skip all tokens between the start and end tag:

<quote> (+)+ </quote> → list of all quotations in the BNC
<mw> (+)+ </mw> → list of all multiword units

- Some useful XML tags in the BNC:

<s> ... </s>	s-unit
<p> ... </p>	paragraph
<u> ... </u>	speaker turn
<head> ... </head>	heading or caption
<quote> ... </quote>	quotation
<item> ... </item>	list item
<hi> ... </hi>	highlighted text
<mw> ... </mw>	multiword unit

Proximity queries

- Special syntax for searching one item within a specified range of another:

kick <<s>> bucket → *kick* and *bucket* in the same sentence
{kick/V} <<s>> bucket_NN1 (can use POS/lemma constraints)
day <<3>> night → *day* and *night* within range of 3 tokens
day <<5<< night → *night* ... *day* (within 5 tokens)
day >>5>> night → *day* ... *night* (within 5 tokens)

- Only the left element ("target") will be highlighted on the result page. The right element is considered as a "constraint" that must be satisfied.
- Multiple constraints can be chained:

{day} <<5>> {month} <<5>> {year}

In this case, *day* must co-occur with *month* as well as *year* in a 5-token window; only *day* will be highlighted on the Query result page.

- Proximity queries can be nested with parentheses:

{waste/V} <<s>> (time <<3>> money)

Here, the verb *waste* must co-occur with *time* as well as *money* in the same sentence; but *time* and *money* must be closer together (within a 3-token window). Again, only instances of *waste* will be highlighted.

- Proximity queries cannot be combined with lexico-grammatical patterns!

List of part-of-speech tags in the BNC (CLAWS-5 tagset)

Tag	Description
AJ0	Adjective (general or positive) (e.g. <i>good, old, beautiful</i>)
AJC	Comparative adjective (e.g. <i>better, older</i>)
AJS	Superlative adjective (e.g. <i>best, oldest</i>)
AT0	Article (e.g. <i>the, a, an, no</i>)
AV0	General adverb: an adverb not subclassified as AVP or AVQ (see below) (e.g. <i>often, well, longer</i> (adv.), <i>furthest</i>)
AVP	Adverb particle (e.g. <i>up, off, out</i>)
AVQ	Wh-adverb (e.g. <i>when, where, how, why, wherever</i>)
CJC	Coordinating conjunction (e.g. <i>and, or, but</i>)
CJS	Subordinating conjunction (e.g. <i>although, when</i>)
CJT	The subordinating conjunction <i>that</i>
CRD	Cardinal number (e.g. <i>one, 3, fifty-five, 3609</i>)
DPS	Possessive determiner-pronoun (e.g. <i>your, their, his</i>)
DT0	General determiner-pronoun: i.e. a determiner-pronoun which is not a DTQ or an AT0.
DTQ	Wh-determiner-pronoun (e.g. <i>which, what, whose, whichever</i>)
EX0	Existential <i>there</i> , i.e. <i>there</i> occurring in the <i>there is...</i> or <i>there are...</i> construction
ITJ	Interjection or other isolate (e.g. <i>oh, yes, mhm, wow</i>)
NN0	Common noun, neutral for number (e.g. <i>aircraft, data, committee</i>)
NN1	Singular common noun (e.g. <i>pencil, goose, time, revelation</i>)
NN2	Plural common noun (e.g. <i>pencils, geese, times, revelations</i>)
NP0	Proper noun (e.g. <i>London, Michael, Mars, IBM</i>)
ORD	Ordinal numeral (e.g. <i>first, sixth, 77th, last</i>) .
PNI	Indefinite pronoun (e.g. <i>none, everything, one</i> (as pronoun), <i>nobody</i>)
PNP	Personal pronoun (e.g. <i>I, you, them, ours</i>)
PNQ	Wh-pronoun (e.g. <i>who, whoever, whom</i>)
PNX	Reflexive pronoun (e.g. <i>myself, yourself, itself, ourselves</i>)
POS	The possessive or genitive marker 's or '
PRF	The preposition <i>of</i>
PRP	Preposition (except <i>of</i>) (e.g. <i>about, at, in, on, with</i>)
PUL	Punctuation: left bracket, i.e. (or [
PUN	Punctuation: general separating mark (. , ! : ; – and ?)
PUQ	Punctuation: quotation mark (' and ")
PUR	Punctuation: right bracket, i.e.) or]
TO0	Infinitive marker <i>to</i>
UNC	Unclassified items which are not appropriately considered as items of the English lexicon.

VBB	The present tense forms of the verb BE (except for <i>is</i> and <i>'s</i>), i.e. <i>am</i> , <i>are</i> , <i>'m</i> , <i>'re</i> and <i>be</i> (subjunctive or imperative)
VBD	The past tense forms of the verb BE: <i>was</i> and <i>were</i>
VBG	The <i>-ing</i> form of the verb BE: <i>being</i>
VBI	The infinitive form of the verb BE: <i>be</i>
VBN	The past participle form of the verb BE: <i>been</i>
VBZ	The <i>-s</i> form of the verb BE: <i>is</i> , <i>'s</i>
VDB	The finite base form of the verb DO: <i>do</i>
VDD	The past tense form of the verb DO: <i>did</i>
VDG	The <i>-ing</i> form of the verb DO: <i>doing</i>
VDI	The infinitive form of the verb DO: <i>do</i>
VDN	The past participle form of the verb DO: <i>done</i>
VDZ	The <i>-s</i> form of the verb DO: <i>does</i> , <i>'s</i>
VHB	The finite base form of the verb HAVE: <i>have</i> , <i>'ve</i>
VHD	The past tense form of the verb HAVE: <i>had</i> , <i>'d</i>
VHG	The <i>-ing</i> form of the verb HAVE: <i>having</i>
VHI	The infinitive form of the verb HAVE: <i>have</i>
VHN	The past participle form of the verb HAVE: <i>had</i>
VHZ	The <i>-s</i> form of the verb HAVE: <i>has</i> , <i>'s</i>
VM0	Modal auxiliary verb (e.g. <i>will</i> , <i>would</i> , <i>can</i> , <i>could</i> , <i>'ll</i> , <i>'d</i>)
VVB	The finite base form of lexical verbs, comprising the indicative, imperative and present subjunctive (e.g. <i>forget</i> , <i>send</i> , <i>live</i> , <i>return</i>)
VVD	The past tense form of lexical verbs (e.g. <i>forgot</i> , <i>sent</i> , <i>lived</i> , <i>returned</i>)
VVG	The <i>-ing</i> form of lexical verbs (e.g. <i>forgetting</i> , <i>sending</i> , <i>living</i> , <i>returning</i>)
VVI	The infinitive form of lexical verbs (e.g. <i>forget</i> , <i>send</i> , <i>live</i> , <i>return</i>)
VVN	The past participle form of lexical verbs (e.g. <i>forgotten</i> , <i>sent</i> , <i>lived</i> , <i>returned</i>)
VVZ	The <i>-s</i> form of lexical verbs (e.g. <i>forgets</i> , <i>sends</i> , <i>lives</i> , <i>returns</i>)
XX0	The negative particle <i>not</i> or <i>n't</i>
ZZ0	Alphabetical symbols (e.g. <i>A</i> , <i>a</i> , <i>B</i> , <i>b</i> , <i>c</i> , <i>d</i>)

References

Hoffmann, Sebastian; Evert, Stefan; Smith, Nicholas; Lee, David; Berglund Prytz, Ylva (2008). *Corpus Linguistics with BNCweb – a Practical Guide*. Volume 6 of *English Corpus Linguistics*. Peter Lang, Frankfurt am Main.