Indri Query Language Quick Reference

TERMS / PROXIMITY

Terms:

- term -- stemmed / normalized term
- "term" -- unstemmed / unnormalized term
- #base64(...) -- converts from base64 -> ascii and then stems and normalizes. useful for including non-parsable terms in a
- #base64quote(...) -- same as #base64 except the the ascii term is unstemmed and unnormalized

Examples:

- dogs
- "NASA"
- #base64(Wyh2Lm4ucC5hLnIucy5hLmIubC5lLild) -- equivalent to query term [(u.n.p.a.r.s.a.b.l.e.)]

Proximity terms:

- #odN(...) -- ordered window -- terms must appear ordered, with at most N-1 terms between each
- #N(...) -- same as #odN
- #uwN(...) unordered window -- all terms must appear within window of length N in any order
- #uw(...) -- unlimited unordered window -- all terms must appear within current context in any order

Examples:

- #1(white house) -- matches "white house" as an exact phrase
- #2(white house) -- matches "white * house" (where * is any word or null)
- #uw2(white house) -- matches "white house" and "house white"

Synonyms:

- #syn(...)
- { ... }
- < ... >

Each of these forms does the same thing. They treat all of the expressions listed as the same term.

Examples:

- #syn(#1(united states) #1(united states of america))
- {dog canine}
- <#1(light bulb) lightbulb>

NOTE: The arguments given to this operator can only be term/proximity expressions.

"Any" operator:

- #any -- used to match extent types

Examples:

Examples:

- #any:PERSON -- matches any occurence of a PERSON extent
- #1(napolean died in #any:DATE) -- matches exact phrases of the form: "napolean died in <date>...</date>"

Field restriction / evaluation:

- expression.f1,,...,fN(c1,...,cN) -- matches when the expression appears in field f1 AND f2 AND ... AND fN and evaluates the expression using the language model defined by the concatenation of fields c1...cN within the document.

Examples:

- dog.title -- matches the term dog appearing in a title extent (uses document language model)
- #1(trevor strohman).person -- matches the phrase "trevor strohman" when it appears in a person extent (uses document la model)
- dog.(title) -- evaluates the term based on the title language model for the document
- #1(trevor strohman).person(header) -- builds a language model from all of the "header" text in the document and evaluates (trevor strohman).person in that context (matches only the exact phrase appearing within a person extent within the header

COMBINING BELIEFS

Belief operators:

- #sum
- #wsum
- #wand (weighted and)
- #or
- #combine
- #weight
- #max
- #not
- #band (boolean and)

Examples:

- #combine(<dog canine> training)
- #combine(#1(white house) <#1(president bush) #1(george bush)>)
- #weight(1.0 #1(white house) 2.0 #1(easter egg hunt))

NOTE: If you are unsure which belief operator to use, it always "safest" to default to using the #combine operator. This operator is choice for queries that combine evidence from simple term/proximity expressions.

Extent retrieval:

- #beliefop[field](query) -- evaluates #beliefop(query) for all extents of type "field" in the document and returns a score for the language model used to evaluate the query is formed from the text of the extent.

Example:

- #combine[sentence](#1(napolean died in #any:DATE)) -- returns a scored list of sentence extents that match the given qu

See the additional extent retrieval documentation for more details. Also, the PowerPoint presentation found **here** discusses more d details and gives further examples.

FILTER OPERATORS

Filter operators:

- #filreq -- filter require
- #filrej -- filter reject

Examples:

- #filreq(sheep #combine(dolly cloning)) -- only consider those documents matching the query "sheep" and rank them accor the query #combine(dolly cloning)
- #filrej(parton #combine(dolly cloning)) -- only consider those documents NOT matching the query "parton" and rank them according to the query #combine(dolly cloning)

NOTE: first argument must always be a term/proximity expression

NUMERIC / DATE FIELD OPERATORS

General numeric operators:

- #less(F N) -- matches numeric field extents of type F if value < N
- #greater(F N) -- matches numeric field extents of type F if value > N
- #between(F N_low N_high) -- matches numeric field extents of type F if N_low < value < N_high
- #equals(F N) -- matches numeric field extents of type F if value == N

Date operators:

- #date:after(D) -- matches numeric "date" extents if date is after D
- #date:before(D) -- matches numeric "date" extents if date is before D
- #date:between(D_low, D_high) -- matches numeric "date" extents if D_low < date < D_high

Accepted date formats:

- 11 january 2004
- 11-JAN-04
- 11-JAN-2004
- January 11 2004
- 01/11/04 (MM/DD/YY)
- 01/11/2004 (MM/DD/YYYY)

Examples:

- #filreq(#less(READINGLEVEL 10) george washington) -- if each document in a collection contained a numeric tag that spe reading level of the document, then this query will only retrieve documents that have a reading level below grade 10 and documents will be ranked according to the query "george washington".
- #combine(european history #date:between(01/01/1800, 01/01/1900)) -- such a query may be constructed to find informa about 19th century european history, as this query will find pages that discuss "european history" and contain 19th century

NOTE: The general numeric operators only work on indexed numeric fields, whereas the date operators are only applicable to a sp indexed numeric field named "date". See the indexing documentation for more on numeric fields.

DOCUMENT PRIORS

Prior

- #prior(NAME) -- creates the document prior specified by the name given

Example:

- #combine(#prior(RECENT) global warming) -- we might create a prior named RECENT to be used to give greater weight to documents that were published more recently.

NOTE: Please see the documentation on priors for more detailed information on how to specify and use priors.