Solr Requirements for Advanced Search

This approach to Blacklight Advanced Search is predicated on the following requirements:

1. Boolean syntax must be supported
2. *Groups* of Solr fields must be searched for individual text boxes in the html search form.
3. The group of Solr fields for a single text box have different weights among themselves (query terms matched in particular Solr fields in the group have different boost factors than query term matches in other Solr fields in the group)

Mapping Text Boxes in Advanced Search Form to Solr query pieces

Each text box (or check box) in the advanced search query form maps to a particular set of Solr fields and boosting values (plain boosting and phrase boosting) to be used in the Solr query. The particular fields and boost values (and phrase fields and boost values) for each text box will be configured on the Solr server, via the "advanced" request handler.

The user query itself is also transformed; see below under "Query Transformation".

For each text box containing user-supplied text, there will be a chunk in the value of q sent to solr that looks like this:

```
_query_:"{!dismax qf=$[qf_name1] pf=$[pf_name1]}[xform-query-
    terms]"
```

Pieces in square brackets will be replaced with specific strings. The bold are all part of the actual string sent to solr: quotes, curly braces, exclamation point and dollar signs are all sent to Solr.

So if the user has typed into two text boxes, and the text boxes are to be combined (AND), then the q param sent to Solr would have this form:

```
q=_query_:"{!dismax qf=$[qf_name1] pf=$[pf_name1]}[xformed-query-
    terms1]"
AND
_query_:"{!dismax qf=$[qf_name2] pf=$[pf_name2]}[xformed-query-
    terms2]"
```

If the user chooses to combine the text boxes with an "OR", the q param sent to Solr would have this form:

```
q=_query_:"{!dismax qf=$[qf_name1] pf=$[pf_name1]}[xformed-query-
    terms1]"
OR
_query_:"{!dismax qf=$[qf_name2] pf=$[pf_name2]}[xformed-query-
    terms2]"
```

At this time, the **qt** argument for the Solr request MUST be set to a standard request handler (in our case, "**advanced**"). Otherwise Solr will always return 0 results.

I will provide a mapping of advanced search form text boxes to particular qf and pf arguments in the config/initializer/stanford_config.rb, along the lines of:

```
config[:advanced] = {
    author => {
      :qf => qf_author,
      :pf => pf_author
    },
    title => {
      :qf => qf_title,
      :pf => pf_title
    },
    pub_info => {
      :qf => qf_publisher,
      :pf => pf_publisher
    }
}
```

Query Transformations

Recognized Boolean Operators

- AND
- OR
- NOT

Boolean operators will always be all chars uppercase (i.e. AND, OR, NOT). Lowercase and mixed case expressions of these words can be treated like any other word. Jennifer explains that this is good, because when people cut and paste titles or whatever, they don't want "and" "And" "or" "Or" "not" "Not" interpreted as Boolean.

The default operator is AND. That is, when there is no operator, AND is implicit.

Term:

Boolean operators work with terms: a word, or a group of words. A term can be:

- characters not separated by whitespace

or

- a "clause" joined by the AND (or plain, which is implicit AND) or OR operator:
    - dog AND cat
    - dog cat
    - dog OR cat

or

- a set of parentheses and the characters inside
    - EACH set of parens is a term
    - what is within each set of parens needs to go through treatment below … but preserve the parens

    - example: (lesbian OR gay) AND (video OR book) has 6 terms:
        1. (lesbian OR gay)
        2. lesbian
        3. gay
        4. (video OR book)
        5. video
        6. book
  - parens can be nested:
    - example: cream AND ((filo AND dough) OR pastry) has 6 terms
        1. cream
        2. ((filo AND dough) OR pastry)
        3. (filo AND dough)
        4. filo
        5. dough
        6. pastry

or

- a "clause" joined by the AND (or plain, which is implicit AND) or OR operator:
  - (lesbian OR gay) (video OR book) has 6 terms:
      1. (lesbian OR gay)
      2. lesbian
      3. gay
      4. (video OR book)
      5. video
      6. book

Nested Parens

EACH set of parens is a term
what is within each set of parens needs to go through treatment below … but preserve the parens and treat the parenthetical expression as a term as well.

```
(dog cat) AND mouse   →  +(+dog +cat) +mouse
(lesbian OR gay) (video OR book) → +(lesbian OR gay) +(video OR book)
(klezmer OR accordion) NOT Russian AND (score OR recording) →
      +(klezmer OR accordion) NOT Russian +(score OR recording)
```

AND

When AND is in user query, add a required symbol ("+") to the preceding and following terms and remove the AND:

```
vladimir AND nabokov   →  +vladimir +nabokov
tom AND lynn AND jessie   →  +tom +lynn +jessie
(dog cat) AND mouse   →  +(+dog +cat) +mouse
(dog AND cat) AND mouse   →  +(+dog +cat) +mouse
((dog cat) AND (mouse bug)) house   →
      +(+(+dog +cat) +(+mouse +bug)) +house
((dog AND cat) (mouse AND bug)) house   →
      +(+(+dog +cat) +(+mouse +bug)) +house
((dog cat) (mouse bug)) AND house   →
      +(+(+dog +cat) +(+mouse +bug)) +house
```

Plain

The default operator is AND, so lack of a Boolean operator implies AND. When a term has no preceding NOT, AND or OR, and has no following AND or OR, add a required symbol ("+"):

```
vladimir nabokov   →  +vladimir +nabokov
black and tan → +black +and +tan
black not white → +black +not +white
(dog cat) mouse   →  +(+dog +cat) +mouse
((dog cat) (mouse bug)) house   →  +(+(+dog +cat) +(+mouse +bug))
+house
```

Combos: AND and Plain

```
tom lynn AND jessie   →  +tom +lynn +jessie
tom AND lynn mcrae   →  +tom +lynn +mcrae
```

OR

OR can be left alone.

```
nabokov OR pushkin   → nabokov OR pushkin
(tom lynn) OR Jessie      →  (tom lynn) OR Jessie
```

Combos: AND and OR

```
(dog AND cat) OR mouse   →  (+dog +cat) OR mouse
(dog OR cat) AND mouse   →  +(dog OR cat) +mouse
tom lynn OR jessie        →  +tom lynn jessie
(Lynn AND Naomi) OR (Stu AND Jessie)   →   (+Lynn +Naomi) OR (+Stu
      +Jessie)
(Lynn OR Naomi) AND (Stu OR Jessie)   →   +(Lynn OR Naomi) +(Stu OR
      Jessie)
```

implicit parens -- in these cases, the explicit OR takes precedence over the "plain" operator, e.g. between "gay" and "video". We must make that precedence explicit with parens.

```
lesbian OR gay video OR book   →   +(lesbian OR gay) +(video OR book)
```

NOT

When there is a ***single term*** for the user query in a text box, and it is preceded by NOT, NOT will move in front of the _query_ piece.

```
NOT cat   →  q=NOT _query_:"{!dismax qf=stuff pf=stuff}cat"
NOT (dog OR cat) →  q=NOT _query_:"{!dismax qf=stuff pf=stuff}(dog OR
     cat)"
NOT (dog AND cat) → q=NOT _query_:"{!dismax qf=stuff pf=stuff}(+dog
     +cat)"
```

When there are ***multiple terms*** in a text box and one or more is preceded by NOT, the NOT should be left alone.

```
cat NOT mouse   →  +cat NOT mouse
NOT mouse cat   →  NOT mouse +cat
cat AND NOT mouse   →   +cat NOT mouse
NOT mouse AND cat  →   NOT mouse +cat
NOT mouse cat dog    →    NOT mouse +cat +dog
dog NOT mouse cat    →    +dog NOT mouse +cat
```

```
dog AND cat NOT mouse   →   +dog +cat NOT mouse
dog OR cat NOT mouse    →    dog OR cat NOT mouse
cat NOT mouse OR dog    →    cat NOT mouse OR dog
```

terms in parens:

```
(dog AND cat) NOT mouse   →  +(+dog +cat) NOT mouse
(dog OR cat) NOT mouse    →  +(dog OR cat) NOT mouse
NOT mouse (dog OR cat)    →  NOT mouse +(dog OR cat)
NOT (dog OR cat) AND mouse    →  NOT (dog OR cat) +mouse
```

Note: ***NOT has precedence***, so if the NOT term is immediately followed by AND or OR, then the term preceding the NOT is the first term of the AND/OR (the following are equivalent due to this rule):

```
cat NOT mouse AND dog  →  +cat NOT mouse +dog
(klezmer OR accordion) NOT Russian AND (score OR recording)
      →    +(klezmer OR accordion) NOT Russian +(score OR recording)
```

Single Term

A single term in a search box, *without any operator*, may be left alone:

```
fred   →   fred
```

single term can be a parenthetical phrase

```
(joe OR fred)   →   (joe OR fred)
((dog cat) AND (mouse bug))   →  (+(+dog +cat) +(+mouse +bug))
```

`NOT blah` → this is *not* treated as a single term, because of the "NOT" See above.