

# AGRIS Search Help

You can use the AGRIS search box to create custom, complex queries by hand. AGRIS search syntax is a subset of Apache Solr query syntax: more precisely, it covers only the Lucene query syntax with some refinements. A query expression is decomposed into a set of unordered clauses of three types:

- a clause can be **mandatory**: for example, to search for only documents containing the word *rice* you should type `+rice`
- a clause can be **prohibited**: for example, all documents except those with *rice* will be retrieved with the query `-rice`
- a clause can be **optional**: `rice`

It's ok for spaces to come between `+` or `-` and the search word.

If the query expression contains at least one mandatory clause, then any optional clause is just optional, but it serves a useful function in scoring documents that match more of them higher. If the query expression does not contain any mandatory clauses, then *at least one* of the optional clauses must match. An alternative syntax uses the **boolean operators** `AND`, `OR`, and `NOT`: the engine translates this syntax in the previous one. Take care about the case of the operators: they should be written in uppercase. When the `AND` operator is used between clauses, then both the left and the right sides of the operand become mandatory, while with the `OR` operator they become optional (redundant, because the default operator is already `OR`):

```
rice AND milk    ==>    +rice +milk
```

```
rice OR milk     ==>    rice milk
```

The `NOT` operator is equivalent to the `-` syntax. Remember that `AND` is equivalent to both sides of the operand being mandatory, and thus a query like the following is equivalent to have each clauses mandatory:

```
rice AND milk OR water AND potato    ==>    +rice +milk +water +potato
```

In order to combine query clauses in some ways, you will need to use **sub-expressions**. This means that you can use parenthesis to compose a query of smaller queries:

```
(rice AND milk) OR (water AND potato)    <==>    (+rice +milk) (+water +potato)
```

Apache Solr extends Lucene query Syntax, supporting **pure negative queries**, but only at the top level query expression. To make this work, you have to take the sub-expression containing only negative clauses, and add the all document query clause `*:*`. Thus, to match all documents containing the word *rice* or that don't contain the word *milk*:

```
rice (-milk **)
```

To have a clause explicitly **search a particular indexed field**, precede the relevant clause with the field's name, and then add a colon. Spaces may be used in-between, but that is generally not done.

```
title:(+rice +milk)
```

The content of the parenthesis is a sub-query, but with the default field being overridden to be the one specified. In AGRIS the default field is a special one containing all the indexed fields. Other fields that can be queried are:

```
ARN, center, centerkey, language, title, alternative, titleSupplement,
citationTitle, author, corporateAuthor, publisher, publicationPlace,
publicationDate, agrovoc, subject, ISSN, ISBN, date, fulltext, abstract
```

To search for two words adjacent to each other and in a specific order you must use double quotes:

```
"water alternative"
```

Related to these queries is the notion of the **term proximity**, aka the slop factor or a near query. If you want to permit the specified words to be separated by no more than say four words in-between, then you could do this:

```
"water alternative"~4
```

A Solr index fundamentally stores analyzed terms (words after lowercasing and other processing), and this is generally what you are searching for. However, you can search on partial words using **wildcard queries**:

- no text analysis is performed on the search word. So you may find yourself lowercasing the text before searching it;
- wildcard process is much slower, because every term ever used in the fields needs to be iterated over to see if it matches the wildcard pattern. Moreover, every matched term is added to an internal query, which could grow to be large, but will fail if it attempts to grow larger than 1024 different terms.

To perform wildcard queries you can use the asterisk to match any number of characters (perhaps none), or the question mark to force a match of any character at that position. For example, to match word that start with `fis` and that have at least two more characters but potentially more:

```
fis??*
```

You can also put the wildcard at the front.

Solr lets you query for **numeric, date, and text ranges**. Thus, to search for all documents submitted to AGRIS from 1990 to 2000 you can type:

```
date:[1990 TO 2000]
```

Solr supports also open-ended range queries by using the asterisk:

```
date:[1990 TO *]
```

An application of range queries is the possibility to perform **existence queries**, i.e. to match all documents that have a value in a field (a sort of NOT NULL). For example, to find all documents that have a link to the fulltext, you can type:

```
fulltext:[* TO *]
```

This can be negated to find documents that do not have a value for the field:

```
-fulltext:[* TO *]
```

The following characters are used by the query syntax:

```
+ - && | | ! ( ) { } [ ] ^ " ~ * ? : \
```

In order to use any of these without their syntactical meaning, you need to **escape** them by preceding \, or to use the double-quotes phrase query.