



Ranking Customization

Based on materials compiled by:

Ze'ev Shalev



Ranking Customization

Based on materials compiled by:
Ze'ev Shalev

Notes:

Primo Ranking Customization

Hello, and welcome to today's lesson entitled "Ranking Customization in Primo".

Like most search engines, Primo aims to present results in descending order of relevance, with the top result being the most relevant. Primo's ranking algorithm is based on several heuristics, some of which are customizable, and can be made to better fit the needs of your institution.

In today's lesson, we will learn how you can affect the ranking of Primo's search results from the Primo back-end.

Copyright Statement

All of the information and material inclusive of text, images, logos, product names is either the property of, or used with permission by Ex Libris Ltd. The information may not be distributed, modified, displayed, reproduced – in whole or in part – without the prior written permission of Ex Libris Ltd.

TRADEMARKS

Ex Libris, the Ex Libris logo, Aleph, SFX, SFXIT, MetaLib, DigiTool, Verde, Primo, Voyager, MetaSearch, MetaIndex and other Ex Libris products and services referenced herein are trademarks of Ex Libris, and may be registered in certain jurisdictions. All other product names, company names, marks and logos referenced may be trademarks of their respective owners.

DISCLAIMER

The information contained in this document is compiled from various sources and provided on an "AS IS" basis for general information purposes only without any representations, conditions or warranties whether express or implied, including any implied warranties of satisfactory quality, completeness, accuracy or fitness for a particular purpose.

Ex Libris, its subsidiaries and related corporations ("Ex Libris Group") disclaim any and all liability for all use of this information, including losses, damages, claims or expenses any person may incur as a result of the use of this information, even if advised of the possibility of such loss or damage.

© Ex Libris Ltd., 2008



Notes:

The Ranking Trail (AKA Today's Agenda)



ExLibris
Primo

Notes:

Today's Agenda.

- We'll start with an introduction to ranking in Primo.
- Then we'll talk about document authorities. Not all documents were created equal, and you can recommend which documents are preferable, based on usage statistics.
- Next, we'll discuss field boosts. For example, if a search term appears in the title field, as opposed to, say the description field, its ranking may be boosted.
- The next topic is recency, how the freshness of a document may affect its ranking.
- We'll talk about synonyms -- how Primo can enrich the search by adding synonymous terms.
- We'll then move on to business logic boosting: how FRBR records and Dedup records are ranked. Here we'll cover institution boosting as well.
- Finally, we'll talk about testing. How you can test the results of your modifications to the ranking, and whether there's been an improvement, or deterioration.

The 10 Commandments for Ranking

-
1. Tf-idf
 2. Phrase/distance boosts
 3. Length norm
 4. Field boost
 5. Synonyms
 6. Implicit/explicit feedback
 7. Recency
 8. Authority factor
 9. Business logic boosting
 10. Blending

ExLibris
Primo

Notes:

As an introduction to ranking in Primo, let's take a look at what we call the **10 commandments for Ranking**:

1. Tf/Idf -- This acronym stands for: "Term frequency" - "Inverse document frequency." It is a well-known ranking-metric.
 - Tf -- means the more frequently a term appears in the document, the higher the document's ranking.
 - Idf -- means that search terms which are very common across all documents gain less weight than rarer search terms.
 - These two are combined to form the Tf-Idf metric, which affects the ranking of the document.
2. Phrase/distance boost -- The closer the search terms are to each other in the document, the higher the ranking of the document. This metric cannot be customized in Primo.
3. Length Norm -- The shorter the field in which the search words are found, the more impact those words have on the document -- increasing its ranking. This metric has a side effect of bubbling up shorter documents.
4. Field boosts -- Not all fields in a document are equal: For

example keywords found in the title and author should have more impact than others. This metric is configurable in the Primo back office.

5. Synonyms – Searching for synonyms (such as adding the term “espionage” for a search on “spying”) can greatly enrich search results, on the one hand; but on the other hand, they can also increase the potential for trash.
6. Implicit/explicit feedback -- Understanding feedback from users to enhance ranking. Explicit feedback is rare; this is when a user will tell you that a particular search result was bad. Implicit feedback consists of analyzing whether the user clicked on the higher results or stored them in her e-shelf, etc. You can use this feedback to identify problems and fix them.
7. Recency -- Can be important when looking for older or newer documents. Generally newer documents should carry more weight.
8. Authority factors -- This metric was made prominent by Google. The search algorithm on its own is not enough, and additional external metrics can help tune search results. One of these metrics is usage. Usage statistics should be taken into account when ranking documents. Not all documents are created equal. For example, a document with the word CNN pasted 20 million times should not supersede CNN.com. Later in the lesson, we will show what you can do in Primo to boost the more relevant documents.
9. Business-logic boosting -- Deduplicated or FRBRized records can be boosted to reflect their numbers. You can also choose to boost documents from your own institution.
10. Blending -- Combining search results from different search engines (such as Primo Local and Primo Central in our case). We'll give an overview of how it's done in Primo, and how you can configure this.

Index-Time vs. Search-Time Boosts

Index-Time Boost

- Computed as follows:
$$\text{Index-time boost} = \text{PNX boost (booster1)} + \text{Dedup boost} + \text{Date boost} + \text{FRBR boosts}$$
- This Boost is always greater than 0.
- Called "Negative" If: $0 < x < 1$
- Called "Positive" If: $x > 1$

Search-Time Boost

- Computed as follows:
$$\text{Search-time boost} = \text{field boosts} \times \text{institution boost}$$

Notes:

Index-Time vs. Search-Time Boosts

Now let's talk about the two types of boosts: Index-time boosts and search-time boosts

Index Time Boost

- These are the boosts a document receives when Primo stores it's data in the index.
- The boosts are all added together to create a single number stored with the document, so we have:

$$\text{Index-Time Boost} = \text{PNX Boost (or "booster1")} + \text{Dedup boost} + \text{Date boost} + \text{FRBR boosts}.$$

- This boost is always greater than zero.
- If the boost is a fraction, we refer to it as a "negative boost", as it will decrease a document's ranking when multiplied by the document score. If a boost is greater than 1, we refer to it as a "positive boost", as it will increase the ranking when it's multiplied by the document score.

Search Time Boost

- This is the boost computed during the search.
- It consists of Field Boosts which reflect search-term hits, and the Institution boost, which are multiplied, and not added.
- This boost is multiplied by the document score (Index-time boost).

How it Works – High Level

- Query
 - **Good**
- Index Document boost = 2.7
- Rank documents matching **Good** then *2.7 for every appearance of **Good** in the matched docs
- Title Boost = 3.5
- Return all Docs matching **Good**
- Docs with **Good** in title multiply doc rank by 3.5 for every occurrence on **Good** in title



Notes:

How it Works

How does it work?

Let's look at an example. First an index-time boost:

Consider the search term: "Good"

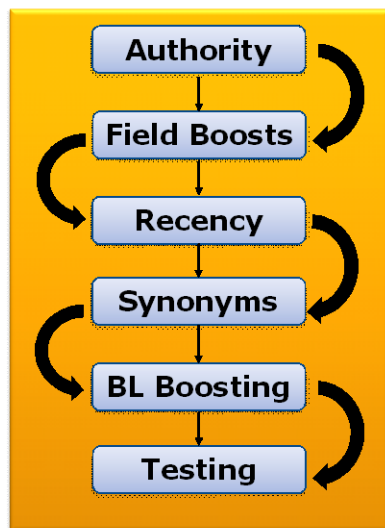
When Primo creates the index it looks at all documents with the word "good" in them. Suppose Primo is now looking at a document with an Index Document Boost of 2.7

The document score will now be multiplied by 2.7 for every appearance of the word "Good" in it. Note: This has potential to reach very high numbers if the Term Frequency is high. So computing the index-time boost must be done carefully.

Now let's look at an example of a search-time boost. The title boost, which is one of the field boosts. Consider again a query on the word "good".

Primo has a default Title Boost of 3.5. Documents scores are multiplied by 3.5 for every occurrence on "Good" in their title

The Ranking Trail – AKA 'The Agenda'



ExLibris
Primo

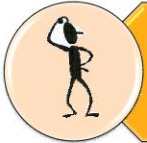
Notes:

Let's move on to authorities.

Authority



Not all documents are equal



Ranking algorithm is smart / dumb



Need to augment ranking algorithm with external knowledge

ExLibris
Primo

Notes:

Document Authority – using click-statistics to assess document quality

Not all documents are created equal. The algorithm cannot on its own determine the importance of the source, based on word-count alone.

On the one-hand, the algorithm is smart: it is designed to get good hits. On the other hand it is dumb: it cannot determine the quality of the document.

The solution is to augment the ranking algorithm with external knowledge, namely usage statistics.

Let's see how this is done in Primo.

Implementing Doc Authority in Primo

Step 1 – The Data

**Export
PNX Click Statistics
for a Specific Date
Range**



Record ID	Add to Eshelf	Getit1	Details
5645464	2	4	12
6546564	6	2	1
3212124	12	3	7

***this excel serves as the input for the plug-in**



Notes:

So how does Primo determine the Authority of a Document?

Let's break it down into 4 steps.

Step 1 -- Collect the data used to determine document quality:

Primo creates a report called "PNX click statistics", which contains some metrics about the popularity of the document:

- How many times was this document added to the e-Shelf?
- How many times was it retrieved?
- How many times did users click to see the details?

Implementing Doc Authority in Primo

Step 1 – The Mechanism

- Use File Splitter to manipulate record content prior to NR
 - Use NR to place a boost in the <Booster1> field based on content in the record
- Write a custom enrichment routine
- **PNX extension plug-in**

**Get smart about
your documents**



Notes:

Implementing document authority in Primo. CONT'D

Step 1 continued -- The Mechanism

This slide states the different ways we can update a document based on usage (that Primo exposes to the library)

There are several ways the usage data can be imported into Primo:

(1) Use File Splitter to manipulate record content prior to Normalization routine. Use Normalization Rules to place a boost in the <Booster1> field based on content in the record.

(2) Write a custom enrichment routine: update the booster field in the PNX record based on the Excel with the number of clicks.

These first 2 mechanisms are problematic: they rely on harvesting data of the PNX record even if nothing changed in the Record itself.

(3) "PNX extension loader"

PNX extension plug-in works best here since it does not require a re-pipe → it is a standalone process that can be run independently and it allows for updates from the external resources only (don't have to manipulate source records so they are piped to get them into the filesplitter / enrichment routines)

Doc Authority in Primo



Notes:

(3) The "PNX extension loader" loads external data directly into a PNX Extension table, which associates data about each of the given PNX records, without changing or reharvesting the record itself. This is the method we will discuss.

Implementing Doc Authority in Primo

Step 2 – The concept

Optional plug-in implementation

- Extract click info per record from the excel
- Calculate a boost based on the clicks
 - (for example: .1 point for details, .5 points for add to eShelf, etc...)
- Subtract boost for older clicks
- Save boost to extension table



Notes:

Step 2- The Concept

Now that we have the click information and the method to use in order to update the boost of records, how do we get the Document Authority boost?

As a simple example to boost the more popular documents: you can implement the following algorithm:

For each record, extract the info from the extension table. Calculate the boost based on clicks, where each click has a different weight, for example:

Add 0.1 points for each click on details tab.

Add 0.5 points for each click on Add to eShelf

etc...

Note that boosting records up is not enough, some records need to be boosted down. A document may be very popular today, but nobody may care about it next week. So because popularity is fleeting, the freshness of the clicks must also be taken into account. With this in mind we provide the following tips:

- Keep the old Excel files too, as they will be needed to de-boost records not clicked on recently. For example, save 4th files back and all records appearing in the 4th and oldest file can receive a negative boost that resets the original boost. A Half-life algorithm works well for de-boosting (boost fast / decay slow)
- The parser receives the Excel as input stream via Back Office configuration for this plug-in (in the publishing subsystem)

Implementing Doc Authority in Primo

Step 3 – Some Coding...

Implement the file splitter interface

```
IFileSplitter  
  
@Override  
public void parse(InputStream is, File file, IRecordSaver saver)  
    throws Exception {  
    RecordData rd = new RecordData(id, "", false);  
    rd.addExtension(new ExtensionData(<EXTENSION NAME>, "1.7"));  
    saver.save(rd);  
}
```



Notes:

Step 3 - Code of the PNX plug-in

Here we can see some lines of code that add document authority information to the PNX extension table.

Implementing Doc Authority in Primo

Step 4

- **Configure**
- **Index**
- **Hot swap**

- **Note this is an index time boost**



Notes:

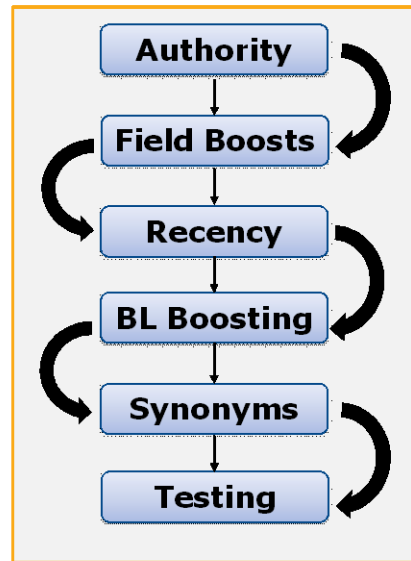
Step (4) for implementing Document Authority is the configuration:

Configure - there are about 4 mapping tables that need to be configured

Index - You have to index records whose boosts have been changed in the PNX extension.

Hot Swap – this should be done without interrupting the system.

The Ranking Trail

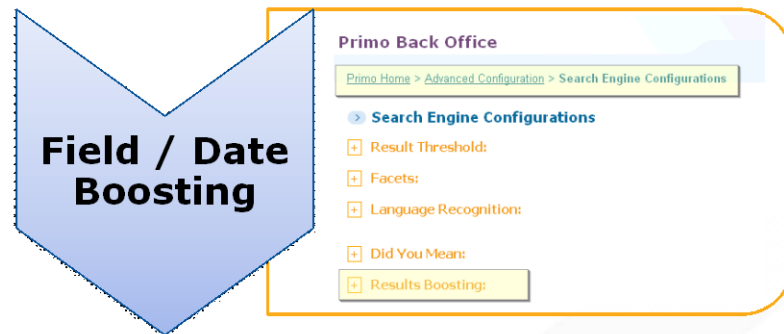


ExLibris
Primo

Notes:

On to Field Boosts.

BO Search Engine configurations



ExLibris
Primo

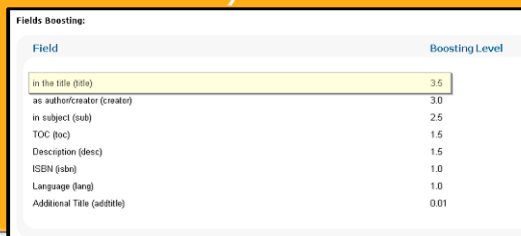
Notes:

Let's talk about Search Engine Configurations in the Primo Back Office.

BO SE Boosting (Query time)

Field boosts

- Boost by N when words from the query match words in boosted field
- Multiply query score by boost (per matched word)



Field	Boosting Level
in title (title)	3.5
as author/creator (creator)	3.0
in subject (sub)	2.5
TOC (toc)	1.5
Description (desc)	1.5
ISBN (isbn)	1.0
Language (lang)	1.0
Additional Title (addtitle)	0.01

exLibris
Primo

Notes:

Field Boosts are a Search-Time boost that can be configured in the Back Office.

Since search-time boosts do not require re-indexing, you can immediately see the results of a change and decide if it's been for the better.

In the Primo Back Office, you can set a different boosting level for each field.

If your Boosting level is N, that means that for every match found in that field, the document score is multiplied by N.

Field boosts give you an opportunity to greatly impact search result behavior. For example, you may decide that a match in the author field is very important. Because if a certain search query matches the name of an author, it makes sense to assume the user is searching an author. So you may want to boost the author field even more than the title field.

BO SE Boosting (Index time)

Date (Recency) boosts

- Boost document by N if it is from a specific year (year range)



Date	Boosting Level	
1990 - 1994	0.9	Delete
2000	1.1	Delete
2009	1.15	Delete
2006	1.04	Delete
2007	1.05	Delete
1995 - 2000	1.0	Delete
1990 - 1999	0.9	Delete
2004	1.02	Delete
2005	1.03	Delete
2001 - 2003	1.01	Delete
2011	3.125	Delete
1900 - 1979	0.6	Delete
2010	2.4	Delete

Create New Date Boosting

Date: 2009 Boosting Level: 0.0 Add

exLibris
Primo

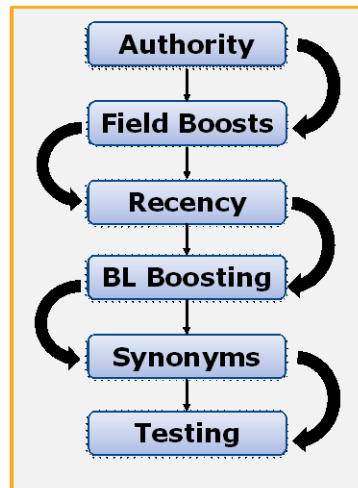
Notes:

Date boosts are an index-time boost that can be configured in the Primo back office. They are added to the document boost.

Configuration of a date boost can be very specific. You can boost by a date range, from year A until year B; or, you can single out a specific year for boosting.

Normally, you will want to give higher boosts to more recent documents, as is shown in the example here.

The Ranking Trail

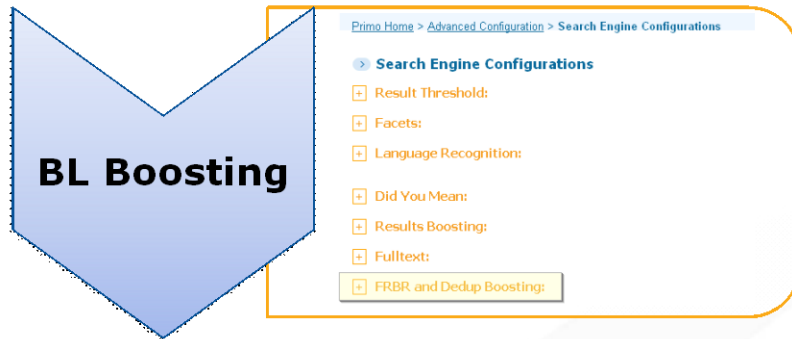


ExLibris
Primo

Notes:

Now let's look at a class of boosts we call "business logic" boosts.

BO Search Engine configurations



ExLibris
Primo

Notes:

SE Boosting (index time / Query time)

FRBR/Dedup/Inst boosting

The screenshot shows the Primo configuration interface for boosting. It includes sections for 'Create New Result Boosting', 'Additional FRBR Boosting' (with 'Availability Boosting' and 'Online Boosting' set to 0.0), 'Dedup Range Boosting' (with a 'Range' input and 'Boosting Level' set to 0.0), and 'Create New Range Boosting' (with 'Range' set to 1.10 and 'Boosting Level' set to 0.0). A central box states: 'For preferred FRBR resource type boosting Give negative boosts → -1 < N < 0'. Below this, a box labeled 'Found in Result Boost Section' points to the 'Institution Boost' table. The table has two columns: 'Field' and 'Value'. The first row is 'Boost for Institution' with a value of 0.5. The word 'Primo' is visible in the bottom right corner of the interface.

Field	Value
Boost for Institution	0.5

Notes:

Primo has three business logic boosts: FRBR boost, Dedup boost and Institution boost.

The first two (FRBR and Dedup) are index-time boosts.

What is a FRBR boost? Primo performs FRBRization on a group of records, but only displays one member of the group. This can lead to a problem because the representative member is the highest ranked of the group, but not necessarily what you want. So Primo helps you fine tune which member of the FRBR group will be the representer.

The first parameter you can configure is the resource type: Primo's out-of-the-box configuration boosts books up and videos down.

The second and third parameter are availability and online. Items that are available in the library can get a higher boost, same with items found online.

The next boost is the Dedup boost. Primo merges identical records into one record that is displayed. A dedup group may represent 200 items or 10 items. This boost allows boosting of larger deduped records.

The third business-logic boost is the "institution boost" and is a search time boost. It works not by boosting up records from your own institution, but rather by boosting down records from outside your institution. This boost is multiplied by the document score. You can make records from outside the institution practically disappear if you set the institution boost to a very small fraction, like 0.001. In that case, the chances are very slim that a search will yield a document from outside the institution.

by default it is 1.0 you should not give a boost greater or equal to -1.0

Institution boosts – this is a negative boost for all institutions that are not yours can be as negative as .001 → will be stronger than exact title matches from other institutions

Query time boosts are multiplied together then multiplied per term match

Summing up Index Time Boosts

Boost	How to change	OTB values
PNX boost Ranking section / booster1	<ul style="list-style-type: none"> Normalization rules File Splitter Enrichment PNX Plugin 	1 (no boost)
Dedup boost Boost records with a lot of members	Search engine configuration	
Date boosts Boost recently published books / articles	Search engine configuration	
FRBR boost Influence the representative FRBR member	Search engine configuration	Video = -0.8 the final boost is 0.2 (negative boost)
		Availability = 0.5 The final boost is 1.5 (positive boost)
		Online = 0.9 The final boost is 1.9 (positive boost)



Notes:

In summary, Let's quickly go over the Index-Time boosts.

The PNX boost, AKA the booster1 field in the PNX record. This field can be populated by the Normalization rules, enrichment routines, and to a certain extent by the file splitter. The PNX extension loader plugin can put in a specific value in the PNX field. The Out-of-The-Box value of this boost is 1, meaning there is no default boosting for any document.

Next we have the dedup boost; this can be configured from the back office. There is no default value for this boost.

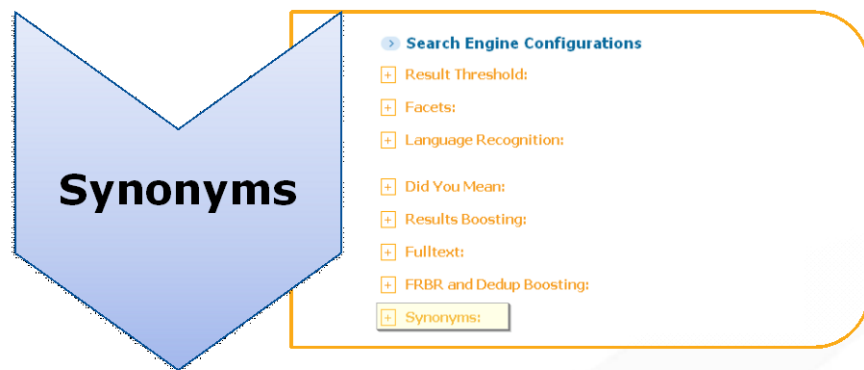
The date boost, boosting recently published articles. This can also be configured in the back office, and by default there is no boost.

FRBR boost, the ability to influence the representing FRBR member. This can be configured from the back office. The default values are as follows.

The boost starts with an initial value of 1, from which the following values are added or subtracted.

video	-0.8 (so when added to 1 it comes to 0.2)
availability	+0.5
online	+0.9

BO Search Engine configurations



ExLibris
Primo

Notes:

Synonyms.

Primo has a fully configurable dictionary file of words and their synonyms. There are two ways synonyms come into play in a Primo search:

(1) Synonyms enrich your search by adding additional search terms found in the synonym file. For example "dog" and "canine".

BO SE Boosting (query time)

Synonyms

1. Synonyms usage is well known and understood
2. Create dominant words using the synonyms mechanism
 - use search term BIRT report

→ Ex. quantum = quantum (very high)

Quantum physics , **Quantum** mechanics, etc...

Field	Value
very high	0.8
high	0.1
normal	0.01
low	0.005
very low	0.0

Notes:

(2) Primo can create dominant words within the search query using the synonyms mechanism.

For example, let's say yours is a science library, and you want certain scientific terms, like "quantum", to have more weight than others.

The trick: in the synonym file, add an entry defining quantum **as a synonym of itself**, like so

quantum = quantum (very high)

Now, if a user searches for "quantum theory", This will cause the word "quantum" to become more important in the query than the word theory. The "very high" in the above example, refers to the quality of the synonym.

Synonyms are a search-time boost.

BO Search Engine configurations

Blending

Blending:

Search Engine: Local Search Engine

☐ Force blending:

Minimum hit rank for combining: High

Combine Location: Top

Number of Results to reward: 0

Constant factor: 1.0



Notes:

Blending.

Blending refers to the combining of results from different search engines. In our case, Primo Central and Primo local.

From the back-end you can configure the boost given to results from the local search engine, and Primo Central. You can fine tune the boosts until you are satisfied that your local results are getting their due representation.

Force Blending

If the default blending does not suit your needs, you can also "force blend" -- which means forcing local or primo central results to get in to the top 10.

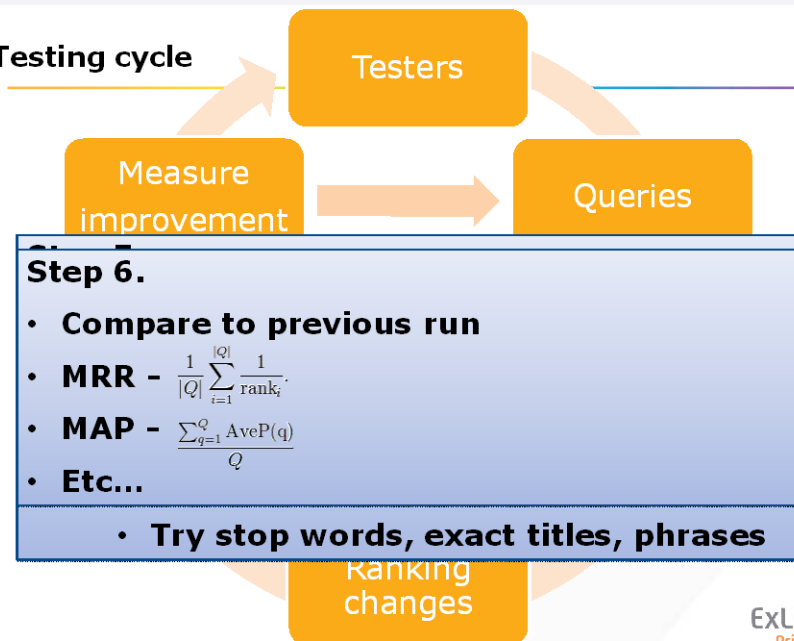
But, you still want these results to be of a certain quality, and not to be there solely on the merit of being local, so Primo lets you specify the minimum hit rank for pushing a result into the top 10. The hit rank qualification is stated simply as High, Medium, and Low (with high meaning a document with a very good score).

Next you decide where the results should be combined: in the top, bottom or middle of the results.

Then you configure the number of results to reward: how many documents get pushed in the top 10.

Primo will push the first result to the top location, and the blending algorithm will decide where to push the next rewarded documents so that they are distributed naturally and not just one after the other.

Testing cycle



Notes:

Testing Cycle.

We've talked about the different ways you can affect ranking, but how do you know if the changes you've made actually work for the better? This is where the testing cycle comes in.

After configuration, you need to test your searches, and then make further improvements based on the tests. Let's outline the testing cycle:

Step 1: Get testers, people to test your system. It is preferable to get people from different fields so you can get a variety of feedback.

Step 2: Each tester should create a list of as many queries as possible. we recommend at least 30 queries, because that increases your chance of seeing how your change affects the query.

However, even if people offer less, that's better than nothing.

Queries should be varied and not simple. Avoid one-word queries like "water".

Try queries with several words or exact phrases, exact titles and authors.

Try queries with stop-words (the, at, for, etc...) Try to be as heterogeneous as possible.

Altogether, around 250 queries should suffice.

Step 3: Each tester should create an excel file for each query with its top 20 results and indicate for each result whether it was a good result or a bad result. No scoring: just good or bad.

Note that the top 3 search-results are the most important in the result set, and you should strive to get good results.

Step 4: Make ranking changes in Primo. If you made changes to index-time results, you will need to re-index

Step 5: Compare to previous run. There are 2 formulas we use to determine the quality of the ranking:

(1) MRR: Mean Reciprocal Rank. This metric measures how close to the top the first good result of the query is.

Where is the first good result of the query? If it was number 5 before, and now number 3 -- we've improved.

(2) MAP: Mean Average Position: This is a function that sums up what percentage the good results are from the entire set of results.

(Note) There are other possible metrics you may wish to research on your own. These are the two that we use.

Summary

A lot of tools to improve ranking in Primo

- Use them, they will make a big impact!

Use external sources

- Listen to what the users want – you have this data in Primo and in your ILS
- Search algorithms will only get you so far

Check your changes

- Even if you only run 10 queries it is still indicative



Notes:

In summary, there are a few final points:

There are many tools that improve ranking in Primo -- use them.

Use external resources. Listen to users feedback, for example using the click statistics data.

And finally: Check your changes. This is perhaps the most important thing. even a small test is better than none.

Thank You!

My.Name@exlibrisgroup.com



Notes:

Thank you for listening to this lesson on Ranking Customization in Primo.