

**FAST Search Engineer's Guide**  
**An Alternate Approach to SBC Absolute Position Boosting**  
*Gaston Gonzalez (gaston.gonzalez [at] lifetech [dot] com)*  
*March 20, 2011*

*FAST ESP's Search Business Center (SBC) provides relevancy management to search engineers and business users through its boosting and blocking capabilities. This whitepaper describes a method for applying SBC-style "Top 10" or "Absolute Position" boosts without the use of SBC.*

### Introduction to the RankTuner

The RankTuner is an out-of-the-box (OOTB) pipeline stage responsible for updating a document's boost information. The RankTuner manipulates three (3) boost-related index fields: *hwboost*, *hwquery* and *hwexcludequery*.

The *hwboost*, or hardwire boost, field contains a numeric value representing the document's static boost. By default this stage assigns 10,000 static boost points to every document plus any value already defined in the *hwboost* field. The static boost is specific to the document and independent of the query.

The *hwquery*, or hardwire query, represents query boosts defined in SBC. The internals of *hwquery* as they relate to absolute position boosts are the subject of this paper.

The *hwexcludequery* field was not analyzed and as such will not be discussed herein.

### hwquery Demystified

Each document may contain *n* number of *hwquery* entries. Usually, there are two (2) entries per boosted query term(s): one for the published search profile and one for the unpublished search profile, assuming that the query boost has been saved and published. Each entry consists of three (3) values: *query*, *boost type* and *boost value*.

The *query* consists of a search profile name, a delimiter (\xc2\xba) and the query term(s).

`booksearchspublished\xc2\xbajava`

The example above illustrates the query format for the query term *java* in a published search profile named *booksearch*.

The *boost type* is a numeric value with supported values of 1 and 2. A boost value of 1 indicates an *Absolute Position* boost and a boost

value of 2 indicates a *Relative Query* boost. The latter is out of scope for this paper.

Given this information, an *Absolute Position* boost is defined as (*query*, [*1-10*], 0) where *query* is of the form `searchprofile\xc2\xbakeyword, [1-10]` is the absolute position for the result between 1 and 10, and the third argument is set to 0. Therefore, if a document has the following *hwquery*:

`(booksearchspublished\xc2\xbajava, 1, 0)`

A search for *java* against the *booksearch* published search profile will result in the document getting returned as the first result.

### Adding Query Terms to Documents

Now that we understand the format of *hwquery*, we need a method for defining the query terms that we wish to boost. One approach, and the approach used by the author, is to attach the query terms to the document. A special field called *hyperkeyword* was created for this purpose. These terms were then parsed by a custom rank tuner stage to programmatically build *hwquery* entries.

Figure 1 Sample FAST Document

```
contentid=0596101872
title=Ajax on Java
author=Steven Douglas Olson
hyperkeyword=java, ajax, olson
description=Ajax on Java shows you how...
```

Given the sample document above, our custom rank tuner produces six (6) entries in *hwquery*.

```
[(booksearchspreview\xc2\xbajava, 1,0),
(booksearchspublished\xc2\xbajava, 1,0),
(booksearchspreview\xc2\xbaajax, 1,0),
(booksearchspublished\xc2\xbaajax, 1,0),
(booksearchspreview\xc2\xbaolson, 1,0),
(booksearchspublished\xc2\xbaolson, 1,0)]
```

Searching for *java*, *ajax* or *olson* results in the document, *Java on Ajax*, being returned as the first document in the result set.

### Caveats

It should be noted that this solution has a few side affects. 1) Documents boosted in this fashion are disconnected from SBC and are not stored in the Cache Manager. 2) Duplicate terms in the hyperkeyword field, across documents, result in the corresponding documents getting ranked in top positions; however, they will compete with each other for relative position. Therefore, if doc1, doc2 and doc3 share the hyperkeyword *lucene*, these documents will appear in the top three results. Their relative

order in the top 3 positions is subject to other relevancy rules.

In terms of the author's goals, caveat #1 is a non-issue as the business requirement was to avoid using SBC and to boost all documents in a given collection by query term to position 1. Secondly, it has been the author's experience that boosting large numbers of queries (over 3,000) using *boostbt* has lead to stability issue on the admin server.

Caveat #2 is also a non-issue as query term collisions still result in the documents getting ranked in top positions. *Figure 2* illustrates this behavior.

Figure 2 Term Collision

Internal Doc ID	Query	1st Result	1st Page	Result #	Total Results	Result Page	Total Pages	Rank Score	
8165713acc52dc1812eca5a56a7ea9f4	FeaturedResults	Gibco Custom Media	true	true	1	156	1	10	10031494
7e108e3314af1cf49e7d870d51428953	FeaturedResults	Genearth	true	true	1	21	1	1	10017388
7e108e3314af1cf49e7d870d51428953	FeaturedResults	custom gene synthesis	true	true	1	19	1	1	10037657
70ac73a569f478713d1f66e95b75a32f4	FeaturedResults	Cell Culture	true	true	1	12178	1	811	10017455
70ac73a569f478713d1f66e95b75a32f4	FeaturedResults	stem cells	true	true	1	1327	1	88	10019384
70ac73a569f478713d1f66e95b75a32f4	FeaturedResults	primary cell	true	true	1	4343	1	289	10016536
421f6696d45e7b2b671376953a0f39b	FeaturedResults	Gibco Media	true	true	1	10264	1	684	10019938
35539eb9e45ea15b20a6b89eabd08f53	FeaturedResults	Vector	true	true	1	5801	1	386	10015447
8190737fac187c01f4fb90d6c8831431	FeaturedResults	human kinome	true	true	1	2	1	0	10024540
454ee611eb067e2994dd197f86f9e23d	FeaturedResults	Oligonucleotides	true	true	1	2001	1	133	10020299
454ee611eb067e2994dd197f86f9e23d	FeaturedResults	custom primer	true	true	1	150	1	10	10024796
454ee611eb067e2994dd197f86f9e23d	FeaturedResults	custom oligo	true	true	1	92	1	6	10026703
454ee611eb067e2994dd197f86f9e23d	FeaturedResults	primer	true	true	1	4327	1	288	10014544
6a533d0e3d7ee74832e6613de52ac611f	FeaturedResults	Vector Designer	true	true	1	13	1	0	10019986
4bce48cafe99f230fae6572d62b5a2fde	FeaturedResults	Peptide	true	true	1	2971	1	198	10017017
61cc064d928af4c7183b74631c03d8e	FeaturedResults	OligoPerfect	true	true	1	2	1	0	10022366
fc052976c5caef72e5f1f61bd8b8d6c7	FeaturedResults	D-lux	true	true	1	21	1	1	10022085
887b8d7dc86c3a227deb3d4a4c2b360a	FeaturedResults	Lux primers	true	true	1	1	14	10024985	
887b8d7dc86c3a227deb3d4a4c2b360a	FeaturedResults	D-lux	false	true	2	1	1	10020880	
9aff3e985ff5c7fcc451f23f20d73cb	FeaturedResults	PlateSelect	true	true	1	1	0	10022366	
9aff3e985ff5c7fcc451f23f20d73cb	FeaturedResults	96-well plate	true	true	1	662	1	44	10024283
de74872f29c614dec9adbfc9c58699f3	FeaturedResults	RNAi	false	true	2	1556	1	103	10020299
de74872f29c614dec9adbfc9c58699f3	FeaturedResults	siRNA	false	true	2	476	1	31	10017125
de74872f29c614dec9adbfc9c58699f3	FeaturedResults	Stealth Select	false	true	2	178	1	11	10026297
797f36c4f5313d3ba700a0d9e8b4eb9c	FeaturedResults	RNAi	true	true	1	1556	1	103	10020299
797f36c4f5313d3ba700a0d9e8b4eb9c	FeaturedResults	siRNA	true	true	1	476	1	31	10017125
797f36c4f5313d3ba700a0d9e8b4eb9c	FeaturedResults	Stealth Select	true	true	1	178	1	11	10026297
4248dcd1eddb1f72240cf4dd30e9a3c82	FeaturedResults	Orf clones	true	true	1	14	1	0	10028753
37af08a5d8b895c33a9576d7456262cc	FeaturedResults	Genes	true	true	1	2184	1	145	10017017
4cf3db726c64e136a76f17eae05f07b	FeaturedResults	Secondary Antibody	true	true	1	2417	1	161	10021477
4cf3db726c64e136a76f17eae05f07b	FeaturedResults	secondary antibodies	true	true	1	2417	1	161	10021477
48cc956008f2358d6cd29597849e24	FeaturedResults	Primary Antibody	true	true	1	4181	1	278	10018984
48cc956008f2358d6cd29597849e24	FeaturedResults	Primary Antibodies	true	true	1	4181	1	278	10018984
93de0c094461a58a2e096e6b22bfb4b6	FeaturedResults	dmem	true	true	1	2451	1	163	10018288
93de0c094461a58a2e096e6b22bfb4b6	FeaturedResults	d-mem	true	true	1	605	1	40	10011000

4 of 35 documents failed relevancy test

### Reverse Engineering Method

Several tools were used during the absolute position analysis. The OOTB FAST tools included: *getfixml* and *doclog*. The *getfixml* utility was used to diff FIXML for the documents under test and *doclog* was used to debug document processing. Given that the vendor did not provide source code for RankTuning.pyc, an open source Python decompiler, *decompyle*, was used to generate source from the byte code. Lastly, a custom tool to benchmark relevancy (i.e. *Figure 2*) was created to visualize the results quickly. *SFERRankalizer* was used initially,

but was quickly replaced by the custom relevancy benchmarking tool given the number of documents under test.

### Conclusion

Programmatic *Absolution Position* or *Top 10* query boosting is technically feasible by simulating the behavior implemented by the OOTB RankTuner. This solution provides an alternate approach for performing automated query boosts at index time without relying on SBC and the Vespa database.