Infrastructure and search

Roman Chyla and the ADS Team

ADS Users Group Meeting, 20-21 Nov. 2019

Two stories

- Lifecycle of a microservice
  - To illustrate ADS development cycle
  - And little bit of architecture (more details in Sergi’s presentation)
- Search algorithm adjustments
  - A problem that has been plaguing ADS for a loooong time
  - Resolved, but not with definitiveness (but good example of a challenge ADS is facing)

The retreat of Russia. Swebach Bernard Edouard (1800-1870) The retreat of Russia in 1812 (oil on canvas 1; 26 x 1; 93) 1838 Museum of Fine Arts Besancon.

https://www.gettyimages.com/detail/news-photo/swebach-bernard-edouard-the-retreat-of-russia-in-1812-1838-news-photo/1048362648

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.

Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite. Paris, le 20 Novembre 1869.

Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. — Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.

Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.

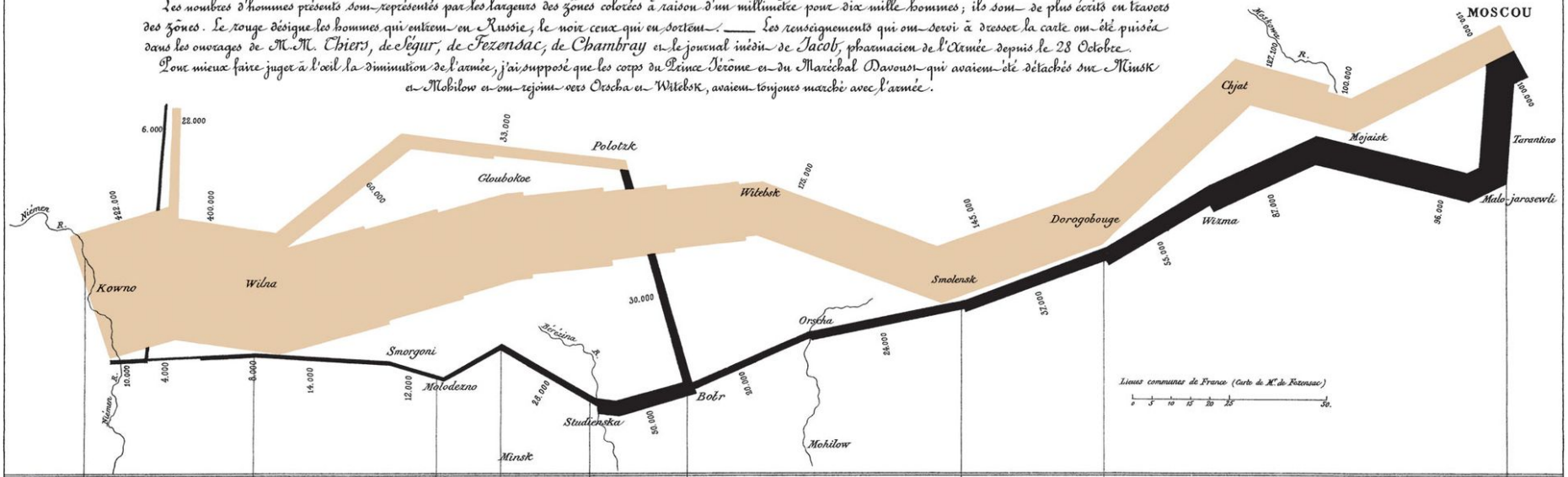
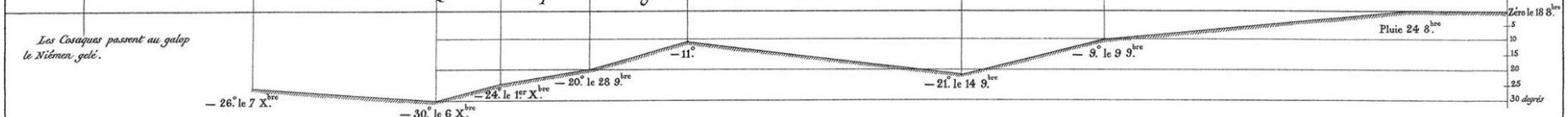


TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.



Autog. par Regnier, 8. Pas. St Marie St Gelin à Paris.

Imp. Lith. Regnier et Dourdet.

First story: lifecycle of a microservice

- Background
  - Multiple iterations (as many as number of devs tackling the problem)
    - 2015 - zip archives (elasticbeanstalk)
    - 2016 - aws api
    - 2017 - kubernetes (manual, piggy-back on eb-deploy)
    - 2018 - keel
    - 2019 - BeeHive + tailor
  - Why is it so hard?
    - It is not an easy problem
    - But it seems un-important (logistics is not “cool”)

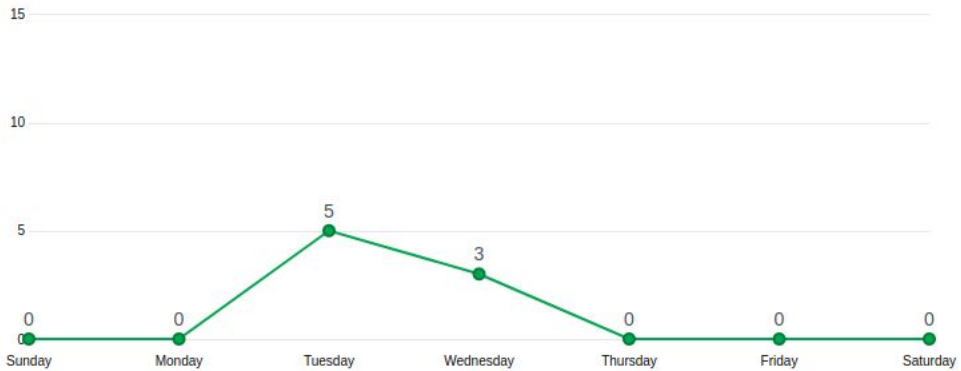
adsabs / **BeeHive** Private

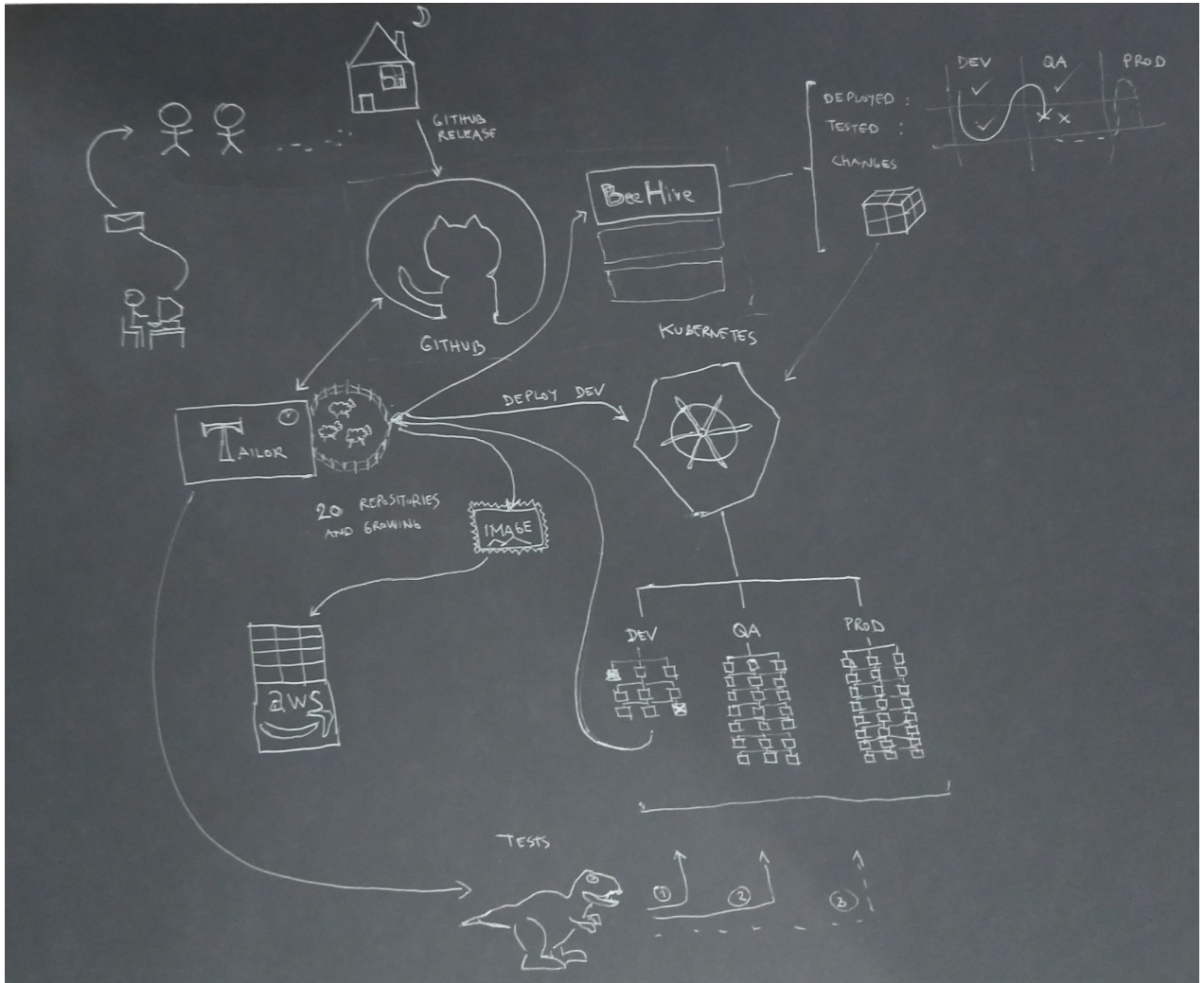
Watch 0 | Star 0 | Fork 6

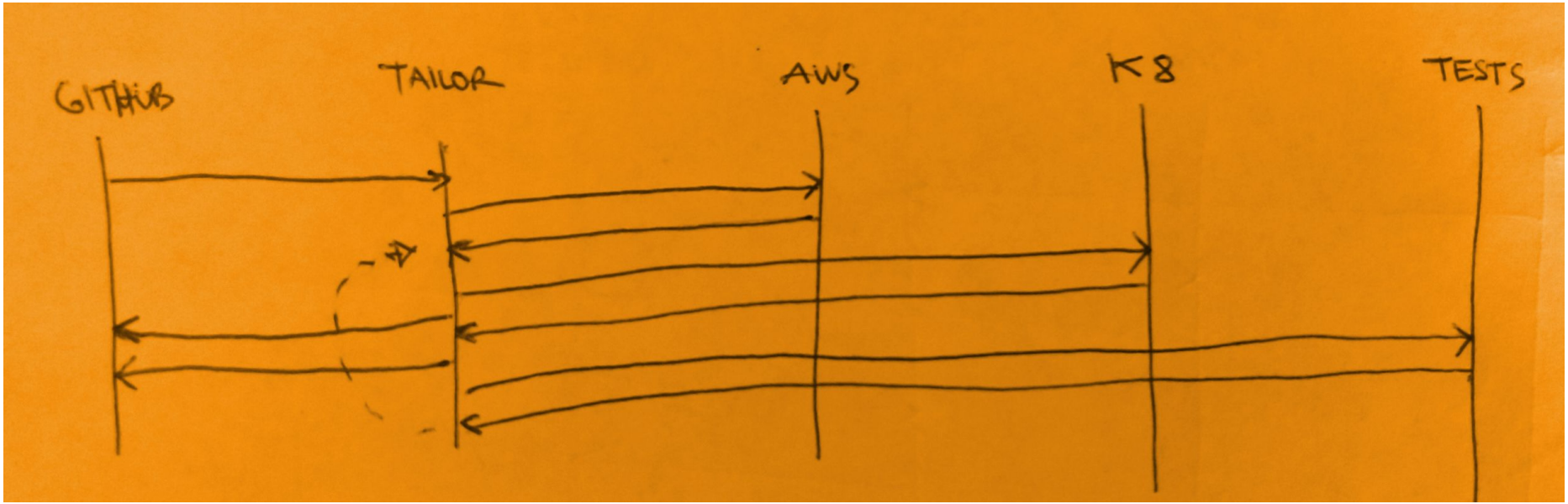
Code | Issues 2 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

- Pulse
- Contributors
- Traffic
- **Commits**
- Code frequency
- Dependency graph
- Network
- Forks

- People




GITHUB RELEASE
BeeHive
DEPLOYED :
TESTED :
CHANGES
DEV
QA
PROD
GITHUB
KUBERNETES
TAILOR
DEPLOY DEV
20 REPOSITORIES AND GROWING
IMAGE
aws
DEV
QA
PROD
TESTS
1
2
3


GITHUB
TAILOR
AWS
K8
TESTS

adsabs / **BeeHive** Private

Watch 0 | Star 0 | Fork 6

Code | Issues 2 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Releases | **Tags**

Draft a new release

Latest release

v0.0.195

a8b941d

v0.0.195

Edit

ads-tailor released this 16 hours ago

built-by: tailor:v0.0.3-50-g79c9165
created: 2019-11-14 00:41:32.230793+00:00
deployed: DEV (2019-11-14T00:41:39.449669+00:00 | success)
tested: DEV (2019-11-14T00:42:15.384814+00:00 | failure)
updated: 2019-11-14 00:42:15.386104+00:00
updated-services: Name=tugboat, release=v2.0.35, image=084981688622.dkr.ecr.us-east-1.amazonaws.com/tailor:tugboat-v2.0.35

Assets 2

- **Source code** (zip)
- **Source code** (tar.gz)

Search

- Significant changes to relevancy computation
  - This was lots of fun
  - Special thanks to Kelly and Alberto
- New algorithm resembles old Classic
  - We don’t know if it is good enough!
    - We like it though
    - And users may not actually care (wonderful example of too much ado about nothing)
  - Examples to illustrate the problem
    - Relevancy in ADS Classic
    - Final score computation in SOLR
    - Picking appropriate weights
    - Avoiding double counting

How Classic ~~sees~~ saw things

- First pass (match/no match) filters out docs
- Score is cumulative (weights of the query parts)

log(1 + norm_cites + norm_reads)

Norm_cites = Age-normalized number of citations

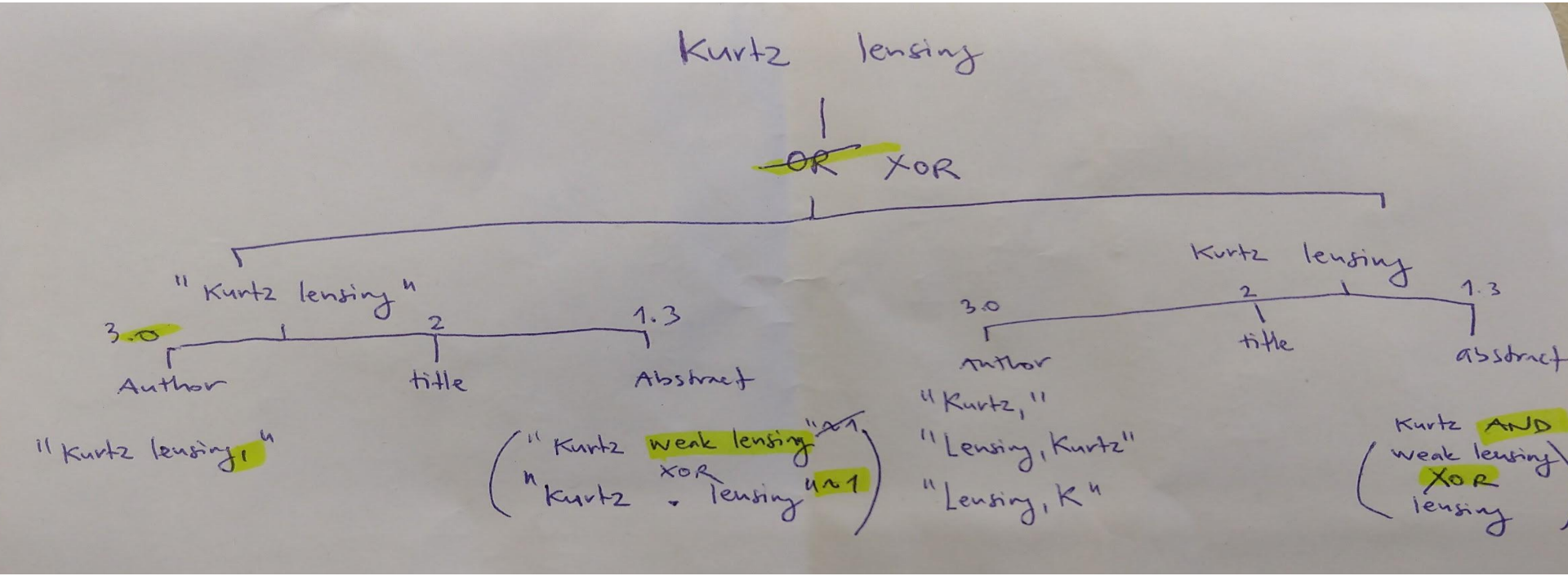
Norm_reads = (cleaned up) reads in the past 90 days

In SOLR, we have this value stored in `cite_read_boost` field (0 < cite_read_boost < 1.0)

Computing final score

score = lucene_score * (cite_read_boost + modifier)

- lucene_score = BM25
- cite_read_boost = see previous slide
- modifier = dubbed “Alberto’s constant” (0.5)
  - Alternative: (1.0-modifier * norm(LS)) + (modifier * cite_read_boost)
- Little bit of arm-twisting still needed
  - Deeply nested query parsing (and hence score computation)
  - But we want this final score to be computed only once


Kurtz lensing
OR
XOR
" Kurtz lensing "
3.0
2
1.3
Author
title
Abstract
" Kurtz lensing, "
(" Kurtz weak lensing "
XOR
" Kurtz lensing ")
Kurtz lensing
3.0
2
1.3
Author
title
abstract
" Kurtz, "
" Lensing, Kurtz "
" Lensing, K "
Kurtz AND
(weak lensing
XOR
lensing)

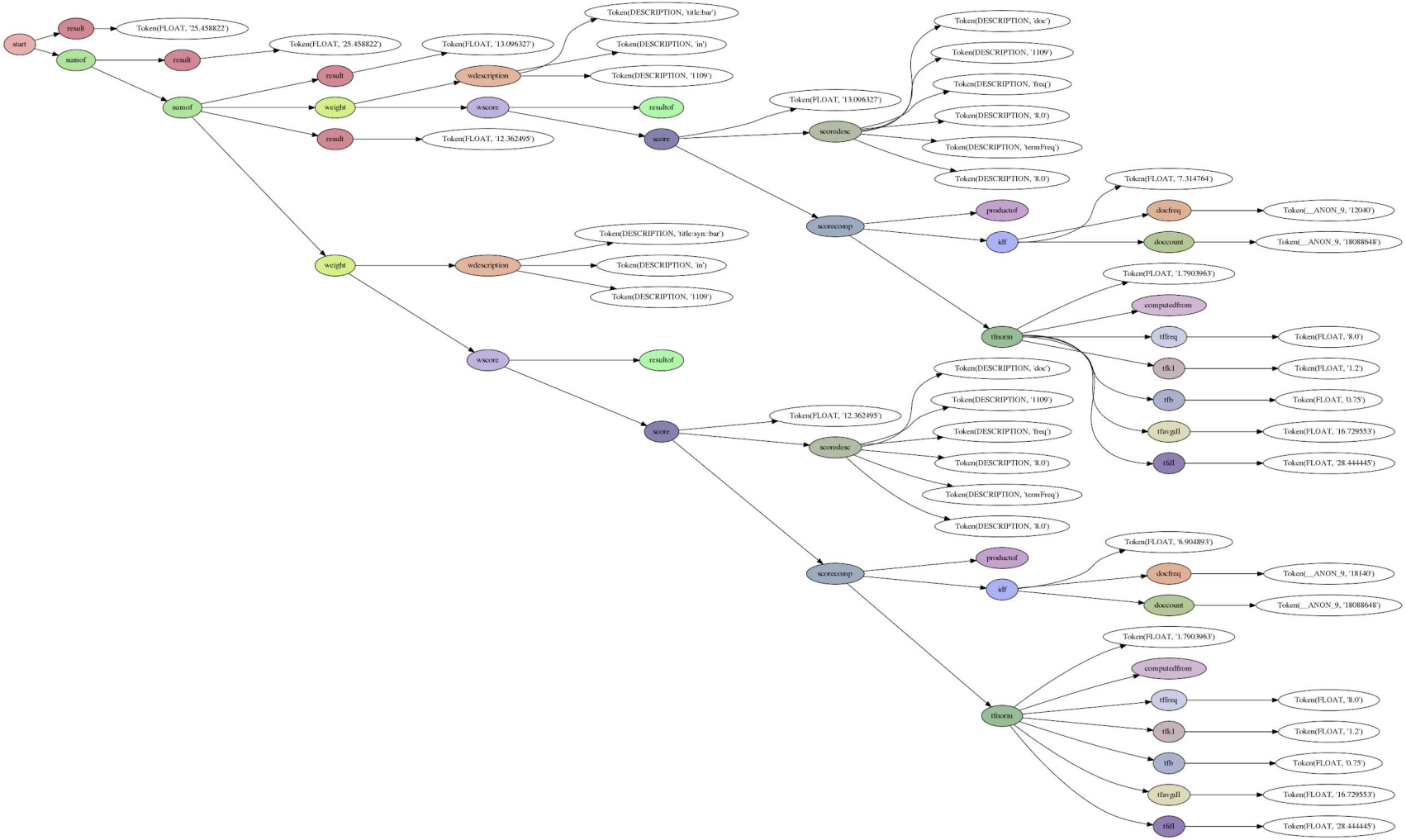
Picking appropriate weights

q=”brown 2000”
Result #2:
2009ApJ...692.1582L

```
46.117043 = custom((+((Synonym(abstract:brown abstract:syn::brown))^1.3 | (author:brown,
author:hanbury, r author:hanbury brown, r author:hanbury brown, robert author:hanbury, robert
author:brown, robert author:brown, r author:brown,*)^2.0 | bibstem:brown |
(first_author:brown, first_author:hanbury, r first_author:hanbury brown, r
first_author:hanbury brown, robert first_author:hanbury, robert first_author:brown, robert
first_author:brown, r first_author:brown,*)^5.0 | identifier:brown | (Synonym(title:brown
title:syn::brown))^1.5 | (year:brown)^2.0) +((abstract:2000)^1.3 | (author:2000,
author:2000,*)^2.0 | bibstem:2000 | (first_author:2000, first_author:2000,*)^5.0 |
identifier:2000 | (title:2000)^1.5 | (year:2000)^2.0)) ((abstract:"(brown syn::brown)
2000")^1.3 | ((author:brown 2000, | author:brown 2000,* | author:2000, brown | author:2000,
brown * | author:2000, b | author:2000, b * | author:2000, | author:2000,*))^2.0 |
bibstem:brown 2000 | ((first_author:brown 2000, | first_author:brown 2000,* |
first_author:2000, brown | first_author:2000, brown * | first_author:2000, b |
first_author:2000, b * | first_author:2000, | first_author:2000,*))^5.0 |
identifier:brown2000 | (title:"(brown syn::brown) 2000")^1.5 | (year:brown2000)^2.0),
```

Score inflation

```
      32.96055 = max of:
        32.96055 = weight(abstract:"(brown syn::brown) 2000" in 166915)
[SchemaSimilarity], result of:
          32.96055 = score(doc=166915,freq=2.0 = phraseFreq=2.0
), product of:
            1.3 = boost
            16.437689 = idf(), sum of:
              6.004394 = idf(docFreq=25422, docCount=10301331)
              5.959437 = idf(docFreq=26591, docCount=10301331)
              4.4738584 = idf(docFreq=117468, docCount=10301331)
            1.5424473 = tfNorm, computed from:
              2.0 = phraseFreq=2.0
              1.2 = parameter k1
              0.75 = parameter b
              185.30257 = avgFieldLength
              113.77778 = fieldLength
```


start
result
Token(FLOAT, '25.458822')
sumof
result
Token(FLOAT, '25.458822')
sumof
result
Token(FLOAT, '13.096327')
weight
wscore
resultof
result
Token(FLOAT, '12.362495')
wdescription
Token(DESCRIPTION, 'title:bar')
Token(DESCRIPTION, 'in')
Token(DESCRIPTION, '1109')
score
Token(FLOAT, '13.096327')
scoredesc
Token(DESCRIPTION, 'doc')
Token(DESCRIPTION, '1109')
Token(DESCRIPTION, 'freq')
Token(DESCRIPTION, '8.0')
Token(DESCRIPTION, 'termFreq')
Token(DESCRIPTION, '8.0')
scorecomp
productof
idf
Token(FLOAT, '7.314764')
docfreq
Token(__ANON_9, '12040')
doccount
Token(__ANON_9, '18088648')
tfnorm
Token(FLOAT, '1.7903963')
computedfrom
tffreq
Token(FLOAT, '8.0')
tfk1
Token(FLOAT, '1.2')
tfb
Token(FLOAT, '0.75')
tfavgdl
Token(FLOAT, '16.729553')
tfdl
Token(FLOAT, '28.444445')
weight
wdescription
Token(DESCRIPTION, 'title:syn::bar')
Token(DESCRIPTION, 'in')
Token(DESCRIPTION, '1109')
wscore
resultof
score
Token(FLOAT, '12.362495')
scoredesc
Token(DESCRIPTION, 'doc')
Token(DESCRIPTION, '1109')
Token(DESCRIPTION, 'freq')
Token(DESCRIPTION, '8.0')
Token(DESCRIPTION, 'termFreq')
Token(DESCRIPTION, '8.0')
scorecomp
productof
idf
Token(FLOAT, '6.904893')
docfreq
Token(__ANON_9, '18140')
doccount
Token(__ANON_9, '18088648')
tfnorm
Token(FLOAT, '1.7903963')
computedfrom
tffreq
Token(FLOAT, '8.0')
tfk1
Token(FLOAT, '1.2')
tfb
Token(FLOAT, '0.75')
tfavgdl
Token(FLOAT, '16.729553')
tfdl
Token(FLOAT, '28.444445')

Summary of changes

- Final score a combination of purely synthetic measures (corpus statistics - BM25) AND paper weight (represented by citations and readership)
- Dozens (if not hundreds) of small adjustments
  - Weights
  - constant vs traditional scores
  - picking strategies for query expansion
  - rewriting author names
  - picking synonyms ….
- But: is that all actually needed?

“I have accomplished much only to accomplish in the end nothing.”

W. Churchill

Crystal hazing (highly subjective view)

- Last mile of the CI (bring the tests to bear weight; automate everything)
- Search - user tracking, time series db (but one is tempted to question the impact; are we invading Italy or France?)
- API - it is handling very large number of requests, but cannot guarantee reliability...
- Kubernetization of back-office components
- Finally (Elephant in the room): new system for curation