# IBM Text Analytics on Apache Spark

**Dimple Bhatia** (dimple@us.ibm.com, @dimpbhatia)
**Sudarshan Thitte** (srthitte@us.ibm.com, @trsudarshan)

Engineering, Text Analytics, IBM

@ Spark Summit 2014

© 2014 IBM Corporation

# IBM Disclaimer on Forward Looking Statements

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

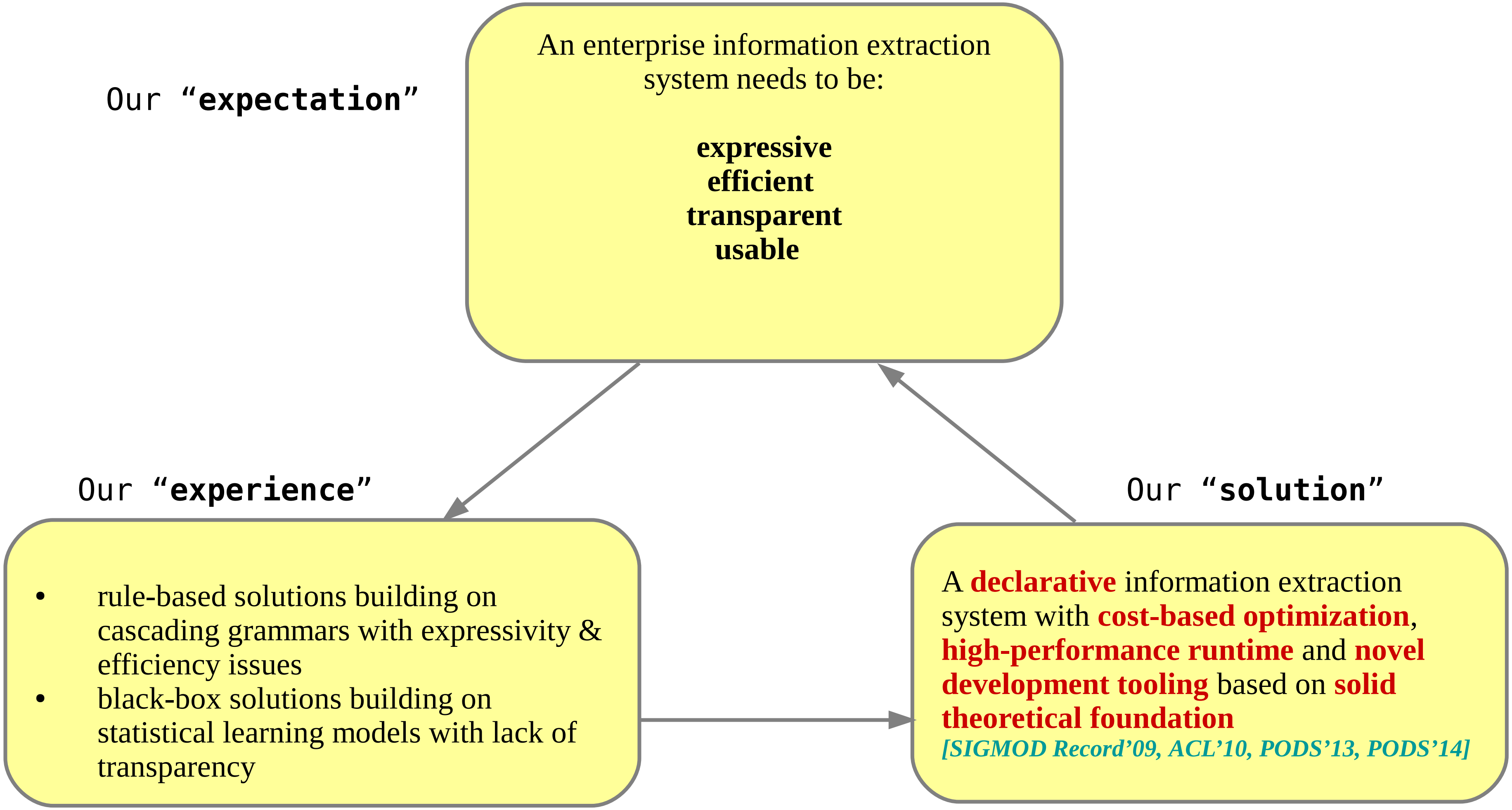
# Agenda

- Motivation
  - IBM Text Analytics → Our expectation, experience, solution

- IBM Text Analytics
  - SystemT → high-performance run-time, uses optimized execution plans
  - Information Extraction ***(IE)*** → deep-parse, lexical semantics, extraction libraries
  - AQL → express lexical semantics as declarative rules using relational algebra
  - Benchmarks → SystemT versus GATE-ANNIE
  - Eclipse & Web based developer tooling → text-analytics life-cycle, map-reduce

- Project ***Sparkle*** - IBM Text Analytics on Apache Spark
  - Spark-Java, Shark-UDTF
  - Future work → Scale, Scala, Tooling, Extractors
  -

# IBM Text Analytics - Motivation



Our “**expectation**”

An enterprise information extraction system needs to be:

**expressive**
**efficient**
**transparent**
**usable**

Our “**experience**”

- rule-based solutions building on cascading grammars with expressivity & efficiency issues
- black-box solutions building on statistical learning models with lack of transparency

Our “**solution**”

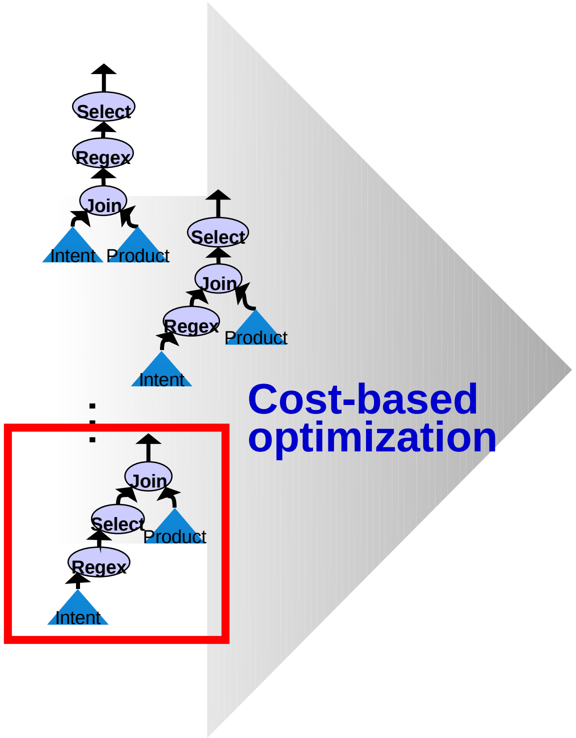
A **declarative** information extraction system with **cost-based optimization**, **high-performance runtime** and **novel development tooling** based on **solid theoretical foundation**

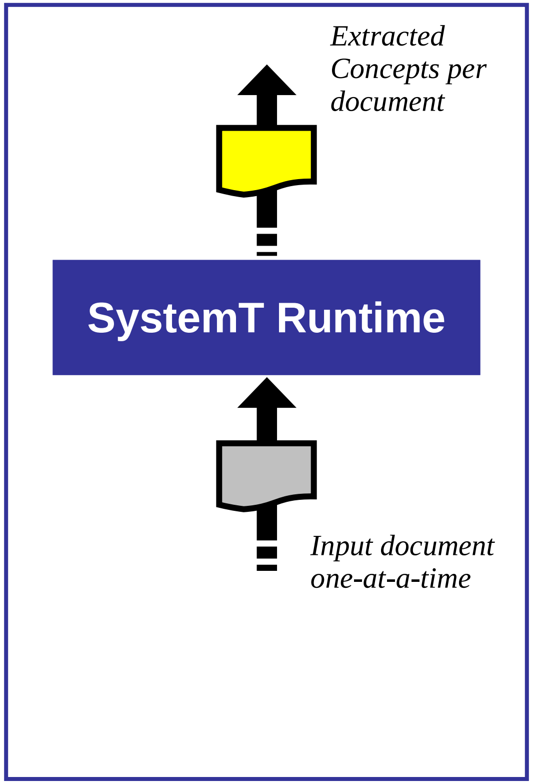
*[SIGMOD Record’09, ACL’10, PODS’13, PODS’14]*

# IBM Text Analytics

## AQL Extractor

```
create view IntentToBuy as
select P.name as product,
       I.clue as strength
from  Intent I, Product P
where
    Follows(I.clue, P.name, 0, 20)
and Not(ContainsRegex(/\b(not)\b/,
    LeftContext(I.clue, 10)));
```





- **Declarative SQL-like language** User specifies tasks in a high-level language, without specifying algorithms for data processing *[SIGMOD Record'09, ACL'10]*
- **High-performance, scalable and embeddable Java runtime Outperforms** state-of-the-art systems *[SIGMOD Record'09, ACL'10]*
- **Modern pattern discovery tools** AQL development using ML & HCI *[EMNLP'08, VLDB'10, ACL'11, CIKM'11, ACL'12, EMNLP'12, CHI'13, SIGMOD'13, ACL'13]*

- **Various optimization strategies to choose across execution plans**

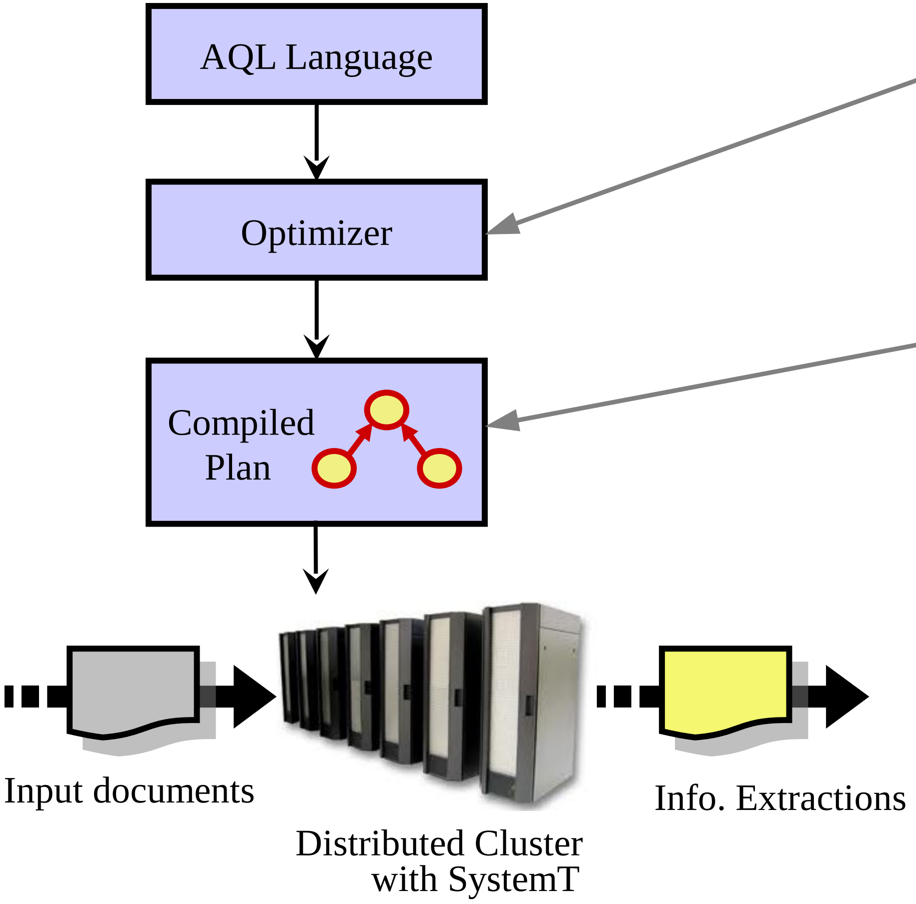
  Cost-based optimization for text-centric operations *[ICDE'08, ICDE'11]*

- **Document-at-a-time**
- **High-throughput**
- **Small memory footprint**

  *[SIGMOD Record'08]*

AQL language exposed via InfoSphere BigInsights and Streams
SystemT Runtime with pre-built extractors ship in 8+ other IBM products

# SystemT – high performance run-time, optimized execution



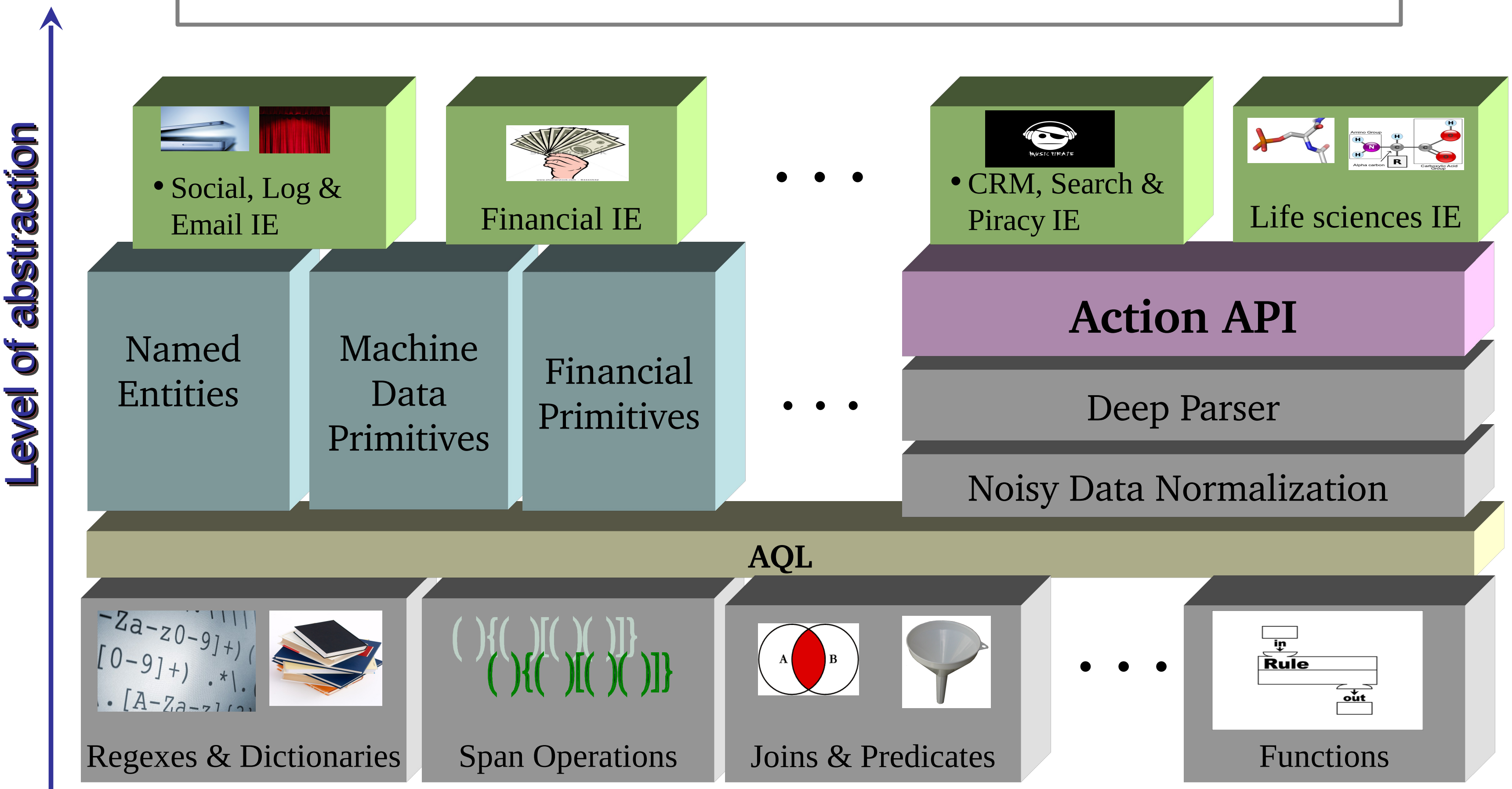
- Multiple ways to execute a given set of AQL statements
- Optimizer chooses a good plan from among alternatives
- Employs multiple techniques
  - AQL rewrite rules
  - Cost-based optimization
  - Global plan rewrite rules

- Extractor plan → graph of *operators*
- *Operator* → a module that performs a specific task, *ex.*: identifying matches of a regex on a string
- Output of one operator → input of another

- Shared Dictionary Matching
- Regular Expression Strength Reduction
- Shared Regular Expression Matching
- Conditional Evaluation

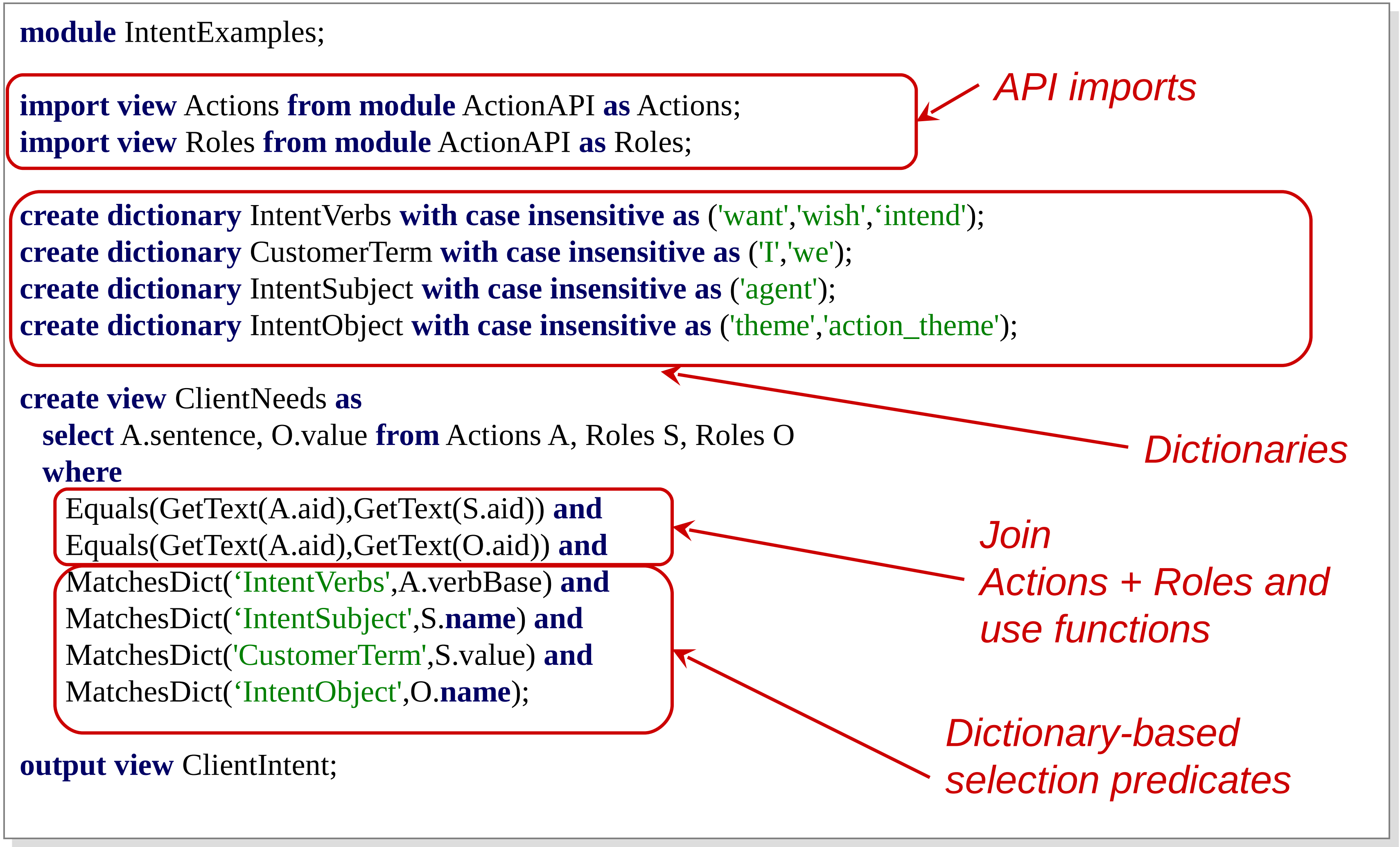
# **Information Extraction** – highlights

*malization, rule-based lexical semantics, algebraic operations over textual spans, extensibility via functions, rich extraction libr*



*Action API, deep-parser & noisy data normalization → work in progress, slated towards a future release of IBM BigInsights*

# AQL – express lexical semantics as declarative rules



```
module IntentExamples;

import view Actions from module ActionAPI as Actions;
import view Roles from module ActionAPI as Roles;

create dictionary IntentVerbs with case insensitive as ('want','wish',‘intend');
create dictionary CustomerTerm with case insensitive as ('I','we');
create dictionary IntentSubject with case insensitive as ('agent');
create dictionary IntentObject with case insensitive as ('theme','action_theme');

create view ClientNeeds as
  select A.sentence, O.value from Actions A, Roles S, Roles O
  where
    Equals(GetText(A.aid),GetText(S.aid)) and
    Equals(GetText(A.aid),GetText(O.aid)) and
    MatchesDict(‘IntentVerbs',A.verbBase) and
    MatchesDict(‘IntentSubject',S.name) and
    MatchesDict('CustomerTerm',S.value) and
    MatchesDict(‘IntentObject',O.name);

output view ClientIntent;
```

*API imports*

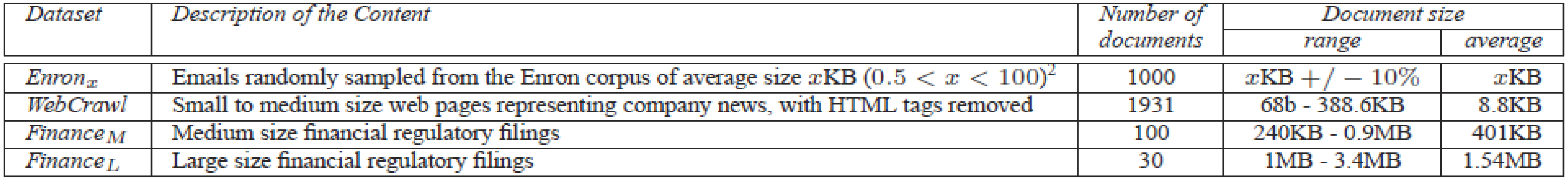
*Dictionaries*

*Join*
*Actions + Roles and use functions*

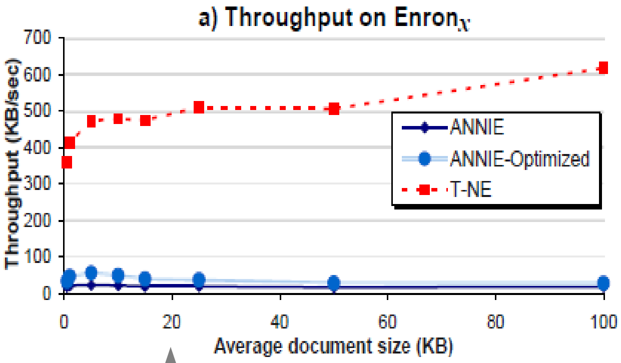
*Dictionary-based selection predicates*

# Benchmarks – SystemT vs GATE-ANNIE+

Table 1: Datasets for performance evaluation.

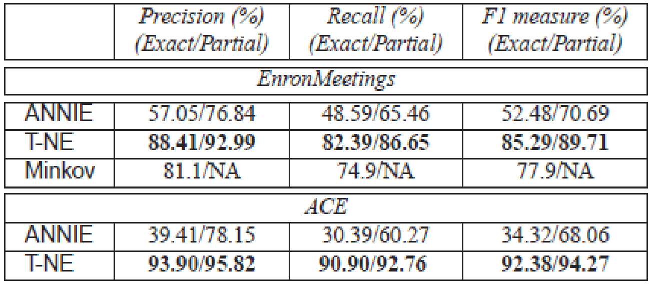


| *Dataset* | *Description of the Content* | *Number of documents* | *Document size* | |
|---|---|---|---|---|
| | | | *range* | *average* |
| *Enron$_x$* | Emails randomly sampled from the Enron corpus of average size $x$KB ($0.5 < x < 100$)$^2$ | 1000 | $x$KB $+/-$ 10% | $x$KB |
| *WebCrawl* | Small to medium size web pages representing company news, with HTML tags removed | 1931 | 68b - 388.6KB | 8.8KB |
| *Finance$_M$* | Medium size financial regulatory filings | 100 | 240KB - 0.9MB | 401KB |
| *Finance$_L$* | Large size financial regulatory filings | 30 | 1MB - 3.4MB | 1.54MB |



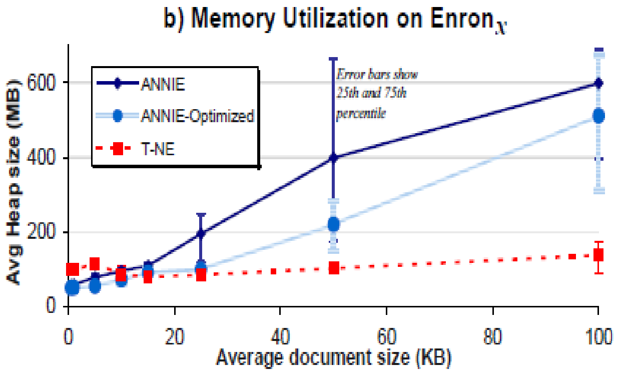
Quality of *Person* entity extraction via AQL is superlative

Table 2: Quality of Person on test datasets.



| | *Precision (%) (Exact/Partial)* | *Recall (%) (Exact/Partial)* | *F1 measure (%) (Exact/Partial)* |
|---|---|---|---|
| *EnronMeetings* | | | |
| ANNIE | 57.05/76.84 | 48.59/65.46 | 52.48/70.69 |
| T-NE | **88.41/92.99** | **82.39/86.65** | **85.29/89.71** |
| Minkov | 81.1/NA | 74.9/NA | 77.9/NA |
| *ACE* | | | |
| ANNIE | 39.41/78.15 | 30.39/60.27 | 34.32/68.06 |
| T-NE | **93.90/95.82** | **90.90/92.76** | **92.38/94.27** |

Runtime Performance* of SystemT is orders of magnitude better

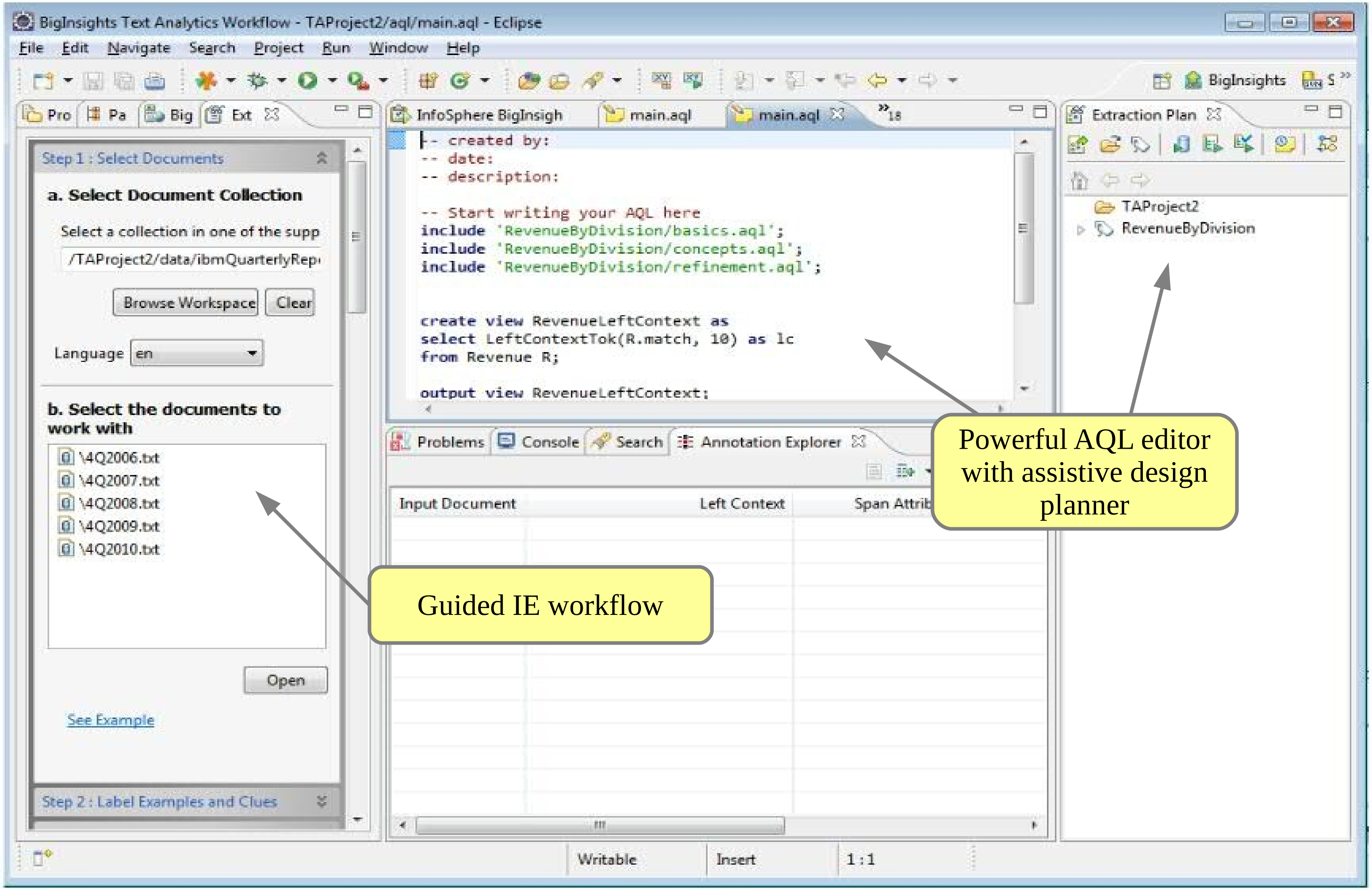


- **T-NE**
  Using AQL & SystemT run-time
- **ANNIE**
  http://gate.ac.uk/sale/tao/splitch6.html#chap:annie
- **ANNIE-Optimized**
  ANNIE with Ontotext Japec transducer
- **Minkov**
  Using E Minkov *[EMNLP’05]*

* as a function of throughput & memory utilization, as seen on a cluster of *2 x 2.4 GHz, 4-core Intel Xeon CPUs with 64GB RAM*

+ GATE-ANNIE is a well known open-source IE system → http://gate.ac.uk/sale/tao/splitch6.html

# **Eclipse IE –** Extraction workflow, AQL editor, extraction design planner



BigInsights Text Analytics Workflow - TAProject2/aql/main.aql - Eclipse

Step 1 : Select Documents

**a. Select Document Collection**

Select a collection in one of the supp

/TAProject2/data/ibmQuarterlyRep

Language: en

**b. Select the documents to work with**

- \4Q2006.txt
- \4Q2007.txt
- \4Q2008.txt
- \4Q2009.txt
- \4Q2010.txt

Step 2 : Label Examples and Clues

```
-- created by:
-- date:
-- description:

-- Start writing your AQL here
include 'RevenueByDivision/basics.aql';
include 'RevenueByDivision/concepts.aql';
include 'RevenueByDivision/refinement.aql';


create view RevenueLeftContext as
select LeftContextTok(R.match, 10) as lc
from Revenue R;

output view RevenueLeftContext;
```

Extraction Plan

- TAProject2
- RevenueByDivision

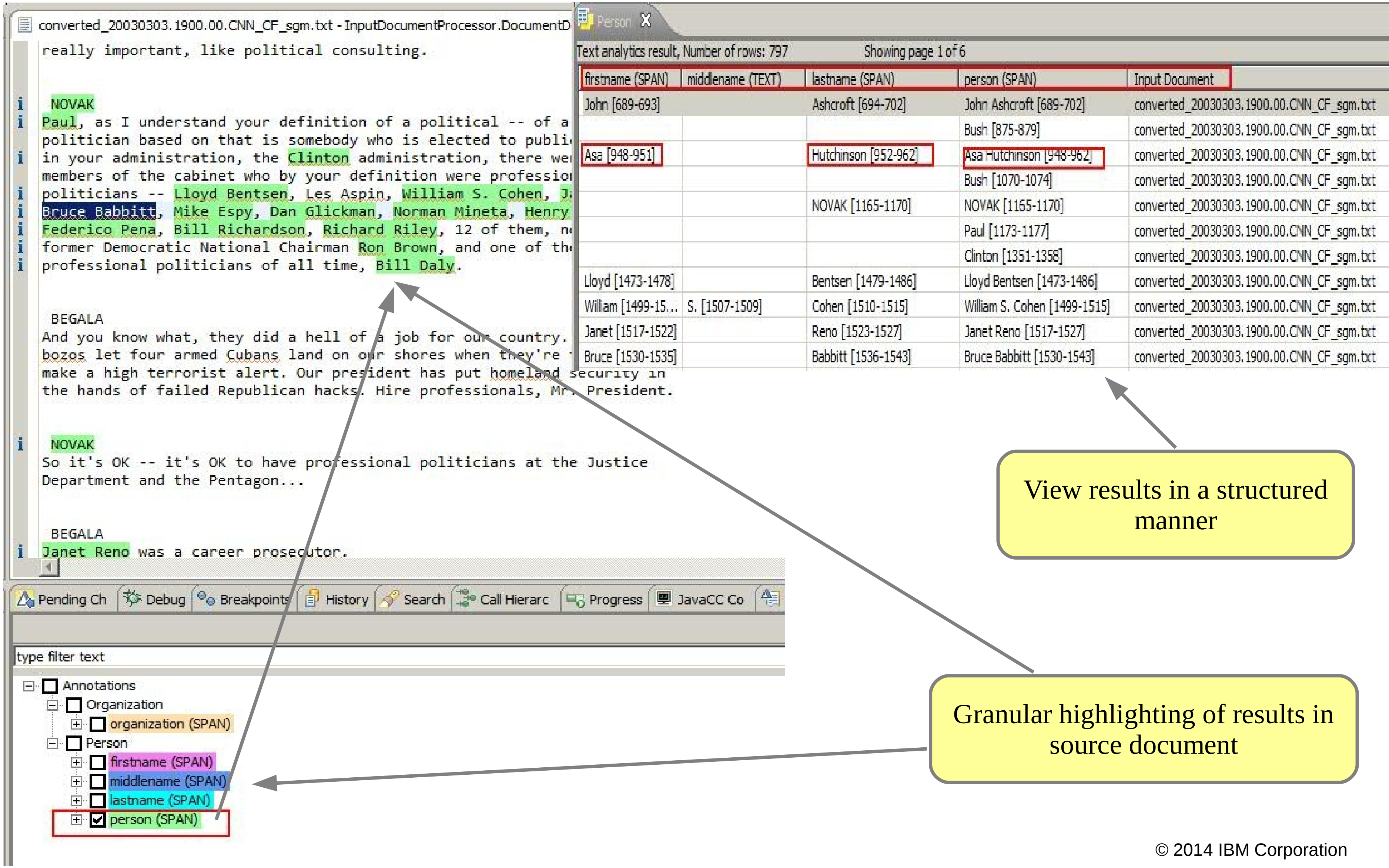
| Input Document | Left Context | Span Attrib |
|---|---|---|

Writable | Insert | 1 : 1

Powerful AQL editor with assistive design planner

Guided IE workflow

# **Eclipse IE –** Result Viewer with granular highlighting



converted_20030303.1900.00.CNN_CF_sgm.txt - InputDocumentProcessor.DocumentD

```
really important, like political consulting.

 NOVAK
Paul, as I understand your definition of a political -- of a
politician based on that is somebody who is elected to publi
in your administration, the Clinton administration, there we
members of the cabinet who by your definition were professio
politicians -- Lloyd Bentsen, Les Aspin, William S. Cohen, Ja
Bruce Babbitt, Mike Espy, Dan Glickman, Norman Mineta, Henry
Federico Pena, Bill Richardson, Richard Riley, 12 of them, n
former Democratic National Chairman Ron Brown, and one of th
professional politicians of all time, Bill Daly.


 BEGALA
And you know what, they did a hell of a job for our country.
bozos let four armed Cubans land on our shores when they're 
make a high terrorist alert. Our president has put homeland security in
the hands of failed Republican hacks. Hire professionals, Mr. President.


 NOVAK
So it's OK -- it's OK to have professional politicians at the Justice
Department and the Pentagon...


 BEGALA
Janet Reno was a career prosecutor.
```

- ☐ Annotations
  - ☐ Organization
    - ☐ organization (SPAN)
  - ☐ Person
    - ☐ firstname (SPAN)
    - ☐ middlename (SPAN)
    - ☐ lastname (SPAN)
    - ☑ person (SPAN)

Person

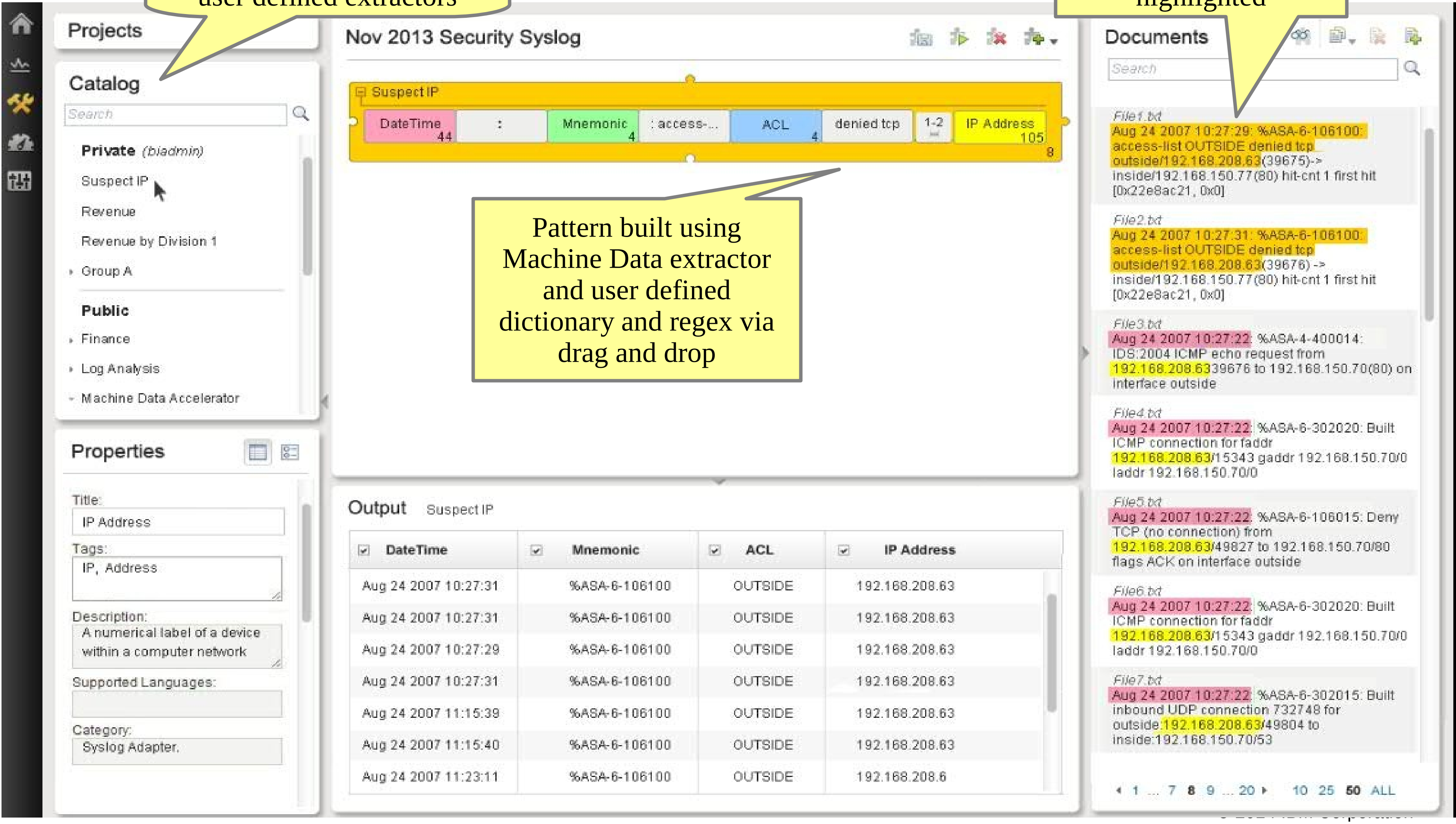
Text analytics result, Number of rows: 797 Showing page 1 of 6

| firstname (SPAN) | middlename (TEXT) | lastname (SPAN) | person (SPAN) | Input Document |
|---|---|---|---|---|
| John [689-693] | | Ashcroft [694-702] | John Ashcroft [689-702] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| | | | Bush [875-879] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| Asa [948-951] | | Hutchinson [952-962] | Asa Hutchinson [948-962] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| | | | Bush [1070-1074] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| | | NOVAK [1165-1170] | NOVAK [1165-1170] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| | | | Paul [1173-1177] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| | | | Clinton [1351-1358] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| Lloyd [1473-1478] | | Bentsen [1479-1486] | Lloyd Bentsen [1473-1486] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| William [1499-15... | S. [1507-1509] | Cohen [1510-1515] | William S. Cohen [1499-1515] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| Janet [1517-1522] | | Reno [1523-1527] | Janet Reno [1517-1527] | converted_20030303.1900.00.CNN_CF_sgm.txt |
| Bruce [1530-1535] | | Babbitt [1536-1543] | Bruce Babbitt [1530-1543] | converted_20030303.1900.00.CNN_CF_sgm.txt |

View results in a structured manner

Granular highlighting of results in source document

# Web-IE (Future release) – Visual extractor development



Title:
IP Address

Tags:
IP, Address

Description:
A numerical label of a device within a computer network

Supported Languages:

Category:
Syslog Adapter.

| ☑ DateTime | ☑ Mnemonic | ☑ ACL | ☑ IP Address |
|---|---|---|---|
| Aug 24 2007 10:27:31 | %ASA-6-106100 | OUTSIDE | 192.168.208.63 |
| Aug 24 2007 10:27:31 | %ASA-6-106100 | OUTSIDE | 192.168.208.63 |
| Aug 24 2007 10:27:29 | %ASA-6-106100 | OUTSIDE | 192.168.208.63 |
| Aug 24 2007 10:27:31 | %ASA-6-106100 | OUTSIDE | 192.168.208.63 |
| Aug 24 2007 11:15:39 | %ASA-6-106100 | OUTSIDE | 192.168.208.63 |
| Aug 24 2007 11:15:40 | %ASA-6-106100 | OUTSIDE | 192.168.208.63 |
| Aug 24 2007 11:23:11 | %ASA-6-106100 | OUTSIDE | 192.168.208.6 |

```
File1.txt
Aug 24 2007 10:27:29: %ASA-6-106100: access-list OUTSIDE denied tcp outside/192.168.208.63(39675)-> inside/192.168.150.77(80) hit-cnt 1 first hit [0x22e8ac21, 0x0]

File2.txt
Aug 24 2007 10:27:31: %ASA-6-106100: access-list OUTSIDE denied tcp outside/192.168.208.63(39676) -> inside/192.168.150.77(80) hit-cnt 1 first hit [0x22e8ac21, 0x0]

File3.txt
Aug 24 2007 10:27:22: %ASA-4-400014: IDS:2004 ICMP echo request from 192.168.208.6339676 to 192.168.150.70(80) on interface outside

File4.txt
Aug 24 2007 10:27:22: %ASA-6-302020: Built ICMP connection for faddr 192.168.208.63/15343 gaddr 192.168.150.70/0 laddr 192.168.150.70/0

File5.txt
Aug 24 2007 10:27:22: %ASA-6-106015: Deny TCP (no connection) from 192.168.208.63/49827 to 192.168.150.70/80 flags ACK on interface outside

File6.txt
Aug 24 2007 10:27:22: %ASA-6-302020: Built ICMP connection for faddr 192.168.208.63/15343 gaddr 192.168.150.70/0 laddr 192.168.150.70/0

File7.txt
Aug 24 2007 10:27:22: %ASA-6-302015: Built inbound UDP connection 732748 for outside:192.168.208.63/49804 to inside:192.168.150.70/53
```

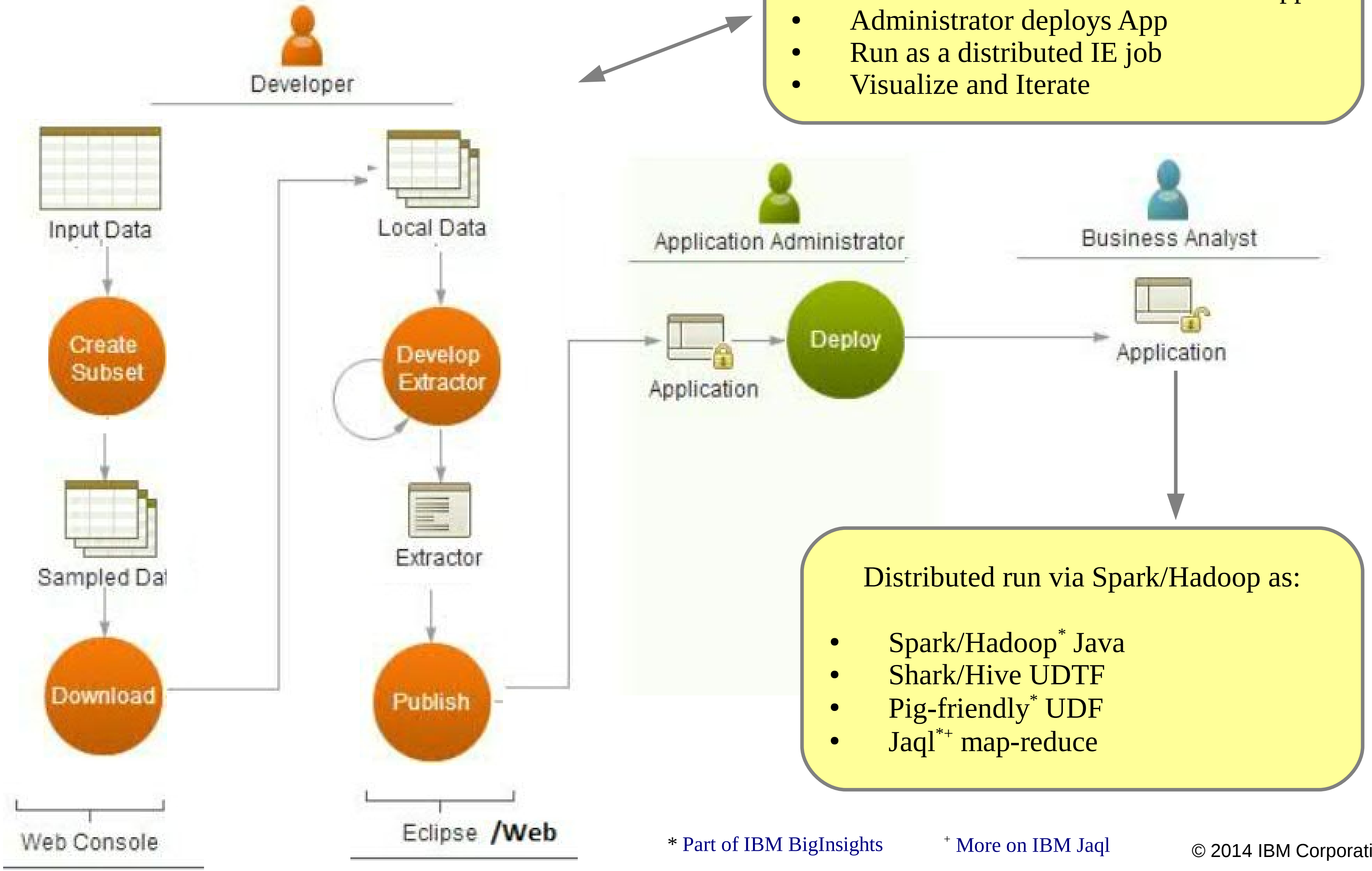
Catalog of primitives and user defined extractors

Extraction Results highlighted

Pattern built using Machine Data extractor and user defined dictionary and regex via drag and drop

# Text Analytics Life-cycle

- Sample/Subset data for training
- Develop IE program/extractor
- Publish to distributed cluster as an App
- Administrator deploys App
- Run as a distributed IE job
- Visualize and Iterate



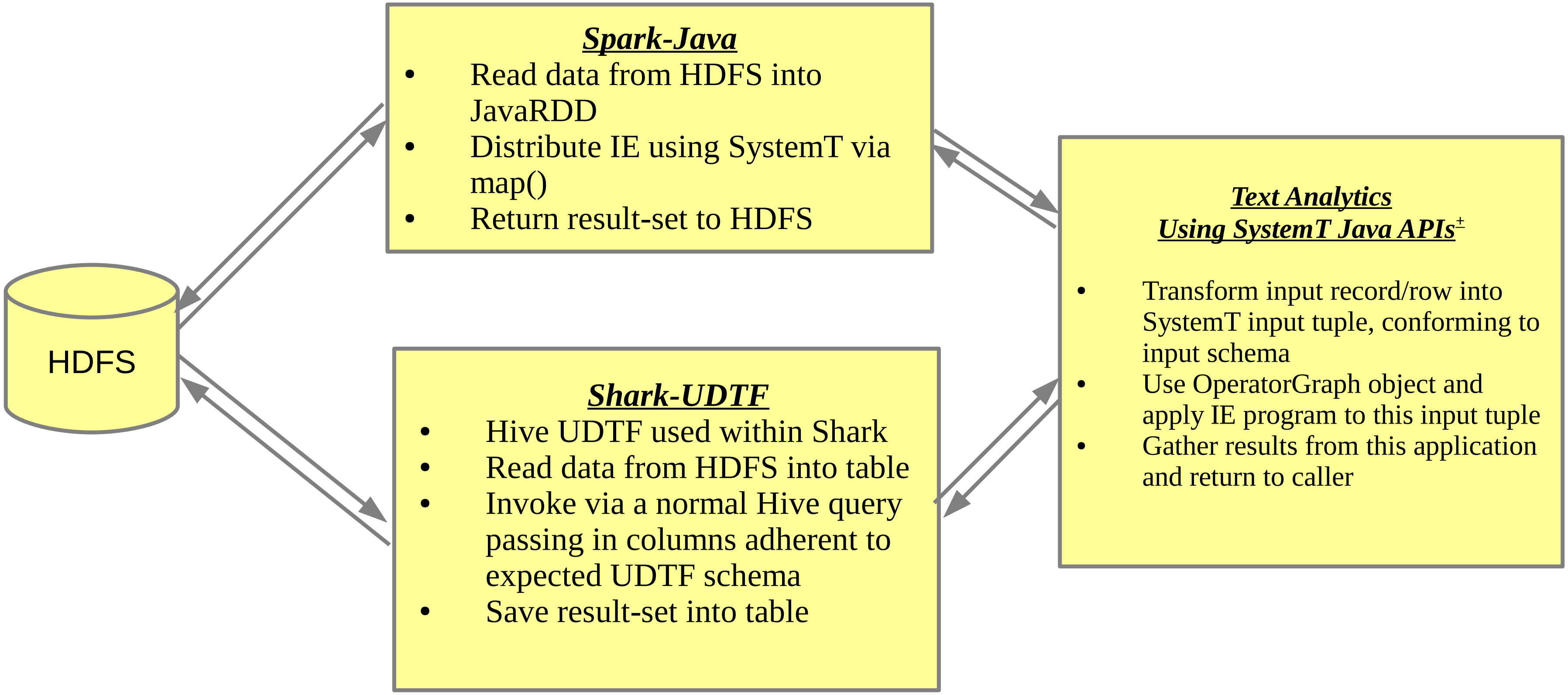
Distributed run via Spark/Hadoop as:

- Spark/Hadoop* Java
- Shark/Hive UDTF
- Pig-friendly* UDF
- Jaql*+ map-reduce

* Part of IBM BigInsights

+ More on IBM Jaql

# *Sparkle* - Text Analytics via Spark Java / Shark UDTF



***Spark-Java***

- Read data from HDFS into JavaRDD
- Distribute IE using SystemT via map()
- Return result-set to HDFS

***Shark-UDTF***

- Hive UDTF used within Shark
- Read data from HDFS into table
- Invoke via a normal Hive query passing in columns adherent to expected UDTF schema
- Save result-set into table

***Text Analytics Using SystemT Java APIs***[+]

- Transform input record/row into SystemT input tuple, conforming to input schema
- Use OperatorGraph object and apply IE program to this input tuple
- Gather results from this application and return to caller

[+] SystemT Java API Tutorial

# *Sparkle* - Future Work

- Stress test current integration with massive data sets and complex IE
- Integrate developer tools with Spark-based IBM text-analytics back-end
- Explore IBM text-analytics as a feature extraction component within large learning-based Spark-analytics pipelines[+]
- Expose IBM Text Analytics to Scala developers[+]

[+] - *long-term*

# References

- 
- Research publications on IBM Text Analytics
    - Contains all research publications around theory, performance & tooling of IBM Text Analytics
- Product documentation on using IBM Text Analytics
    - Documentation regarding our text-analytics technology – its components, usage, tutorials etc.
    - 
- Reference documentation on IBM Text Analytics
    - Official reference documentation for AQL and SystemT's Javadocs
-

# We're hiring !! 😁

Would you like to do **MORE** ?

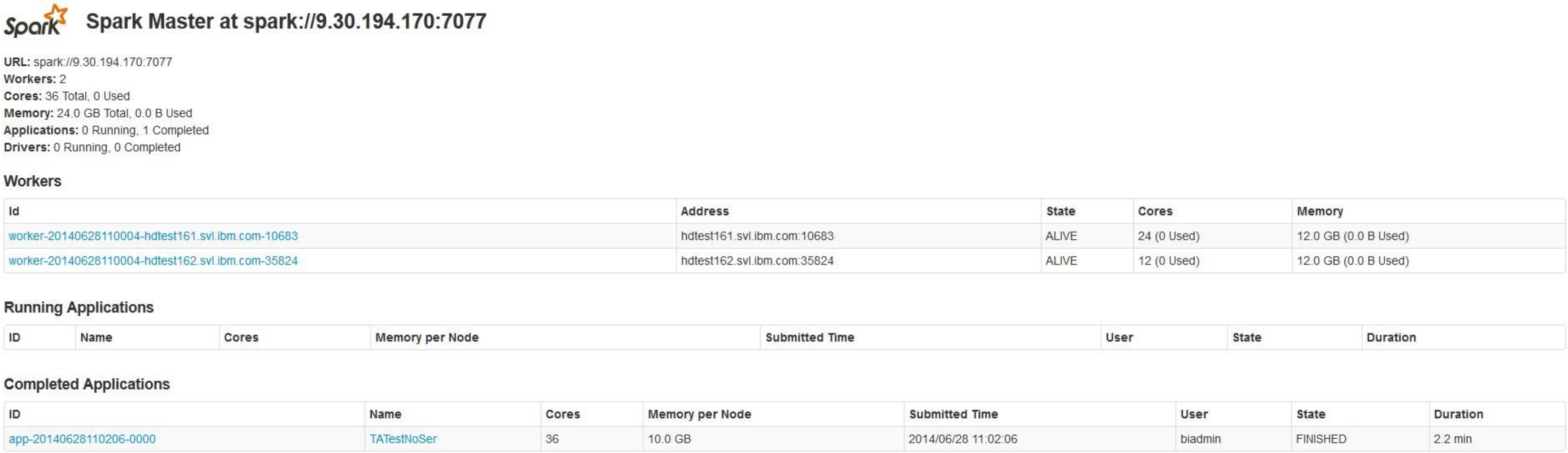
**M**assive-scale data analytics
**O**pen-source commitment
**R**esilient distributed systems
**E**fficient query languages

**If so, talk to us!**

**Dimple** (dimple@us.ibm.com)
**Sudarshan** (srthitte@us.ibm.com)

# BACKUP CONTENT

# Named Entity Extraction via Spark Java



## Spark Master at spark://9.30.194.170:7077

**URL:** spark://9.30.194.170:7077
**Workers:** 2
**Cores:** 36 Total, 0 Used
**Memory:** 24.0 GB Total, 0.0 B Used
**Applications:** 0 Running, 1 Completed
**Drivers:** 0 Running, 0 Completed

### Workers

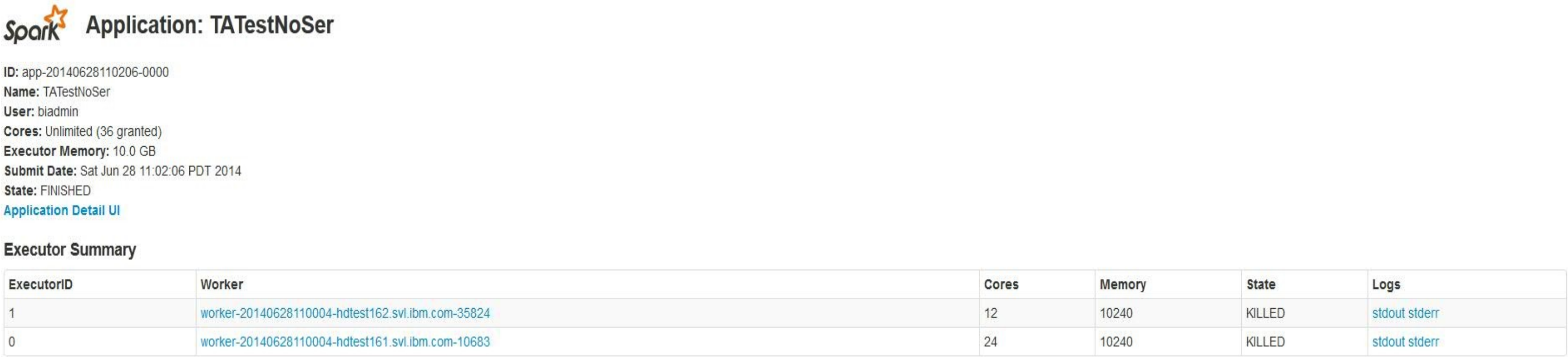
| Id | Address | State | Cores | Memory |
|---|---|---|---|---|
| worker-20140628110004-hdtest161.svl.ibm.com-10683 | hdtest161.svl.ibm.com:10683 | ALIVE | 24 (0 Used) | 12.0 GB (0.0 B Used) |
| worker-20140628110004-hdtest162.svl.ibm.com-35824 | hdtest162.svl.ibm.com:35824 | ALIVE | 12 (0 Used) | 12.0 GB (0.0 B Used) |

### Running Applications

| ID | Name | Cores | Memory per Node | Submitted Time | User | State | Duration |
|---|---|---|---|---|---|---|---|

### Completed Applications

| ID | Name | Cores | Memory per Node | Submitted Time | User | State | Duration |
|---|---|---|---|---|---|---|---|
| app-20140628110206-0000 | TATestNoSer | 36 | 10.0 GB | 2014/06/28 11:02:06 | biadmin | FINISHED | 2.2 min |



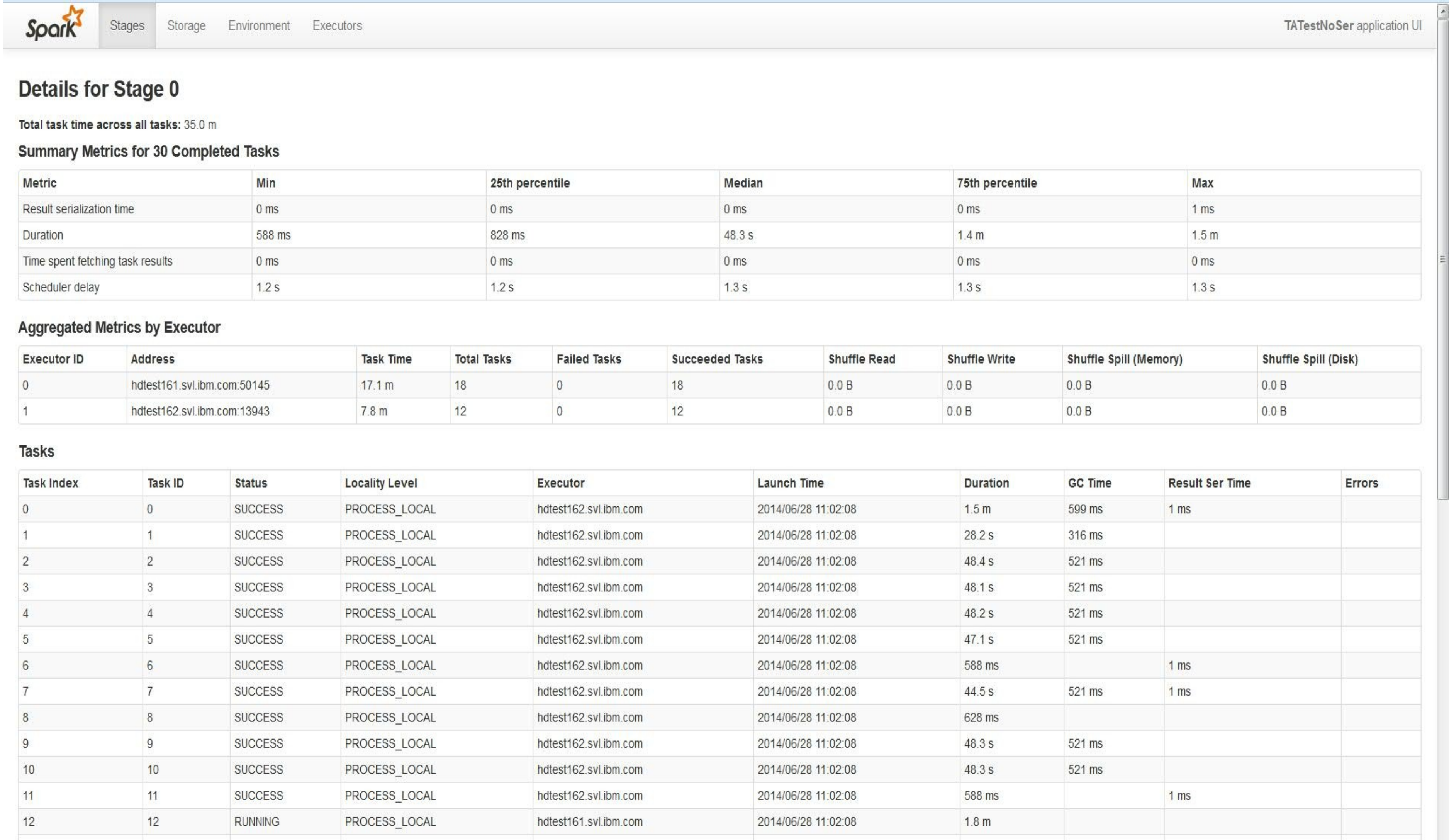
## Application: TATestNoSer

**ID:** app-20140628110206-0000
**Name:** TATestNoSer
**User:** biadmin
**Cores:** Unlimited (36 granted)
**Executor Memory:** 10.0 GB
**Submit Date:** Sat Jun 28 11:02:06 PDT 2014
**State:** FINISHED
**Application Detail UI**

### Executor Summary

| ExecutorID | Worker | Cores | Memory | State | Logs |
|---|---|---|---|---|---|
| 1 | worker-20140628110004-hdtest162.svl.ibm.com-35824 | 12 | 10240 | KILLED | stdout stderr |
| 0 | worker-20140628110004-hdtest161.svl.ibm.com-10683 | 24 | 10240 | KILLED | stdout stderr |

# Named Entity Extraction via Spark Java - Details



## Details for Stage 0

**Total task time across all tasks:** 35.0 m

### Summary Metrics for 30 Completed Tasks

| Metric | Min | 25th percentile | Median | 75th percentile | Max |
|---|---|---|---|---|---|
| Result serialization time | 0 ms | 0 ms | 0 ms | 0 ms | 1 ms |
| Duration | 588 ms | 828 ms | 48.3 s | 1.4 m | 1.5 m |
| Time spent fetching task results | 0 ms | 0 ms | 0 ms | 0 ms | 0 ms |
| Scheduler delay | 1.2 s | 1.2 s | 1.3 s | 1.3 s | 1.3 s |

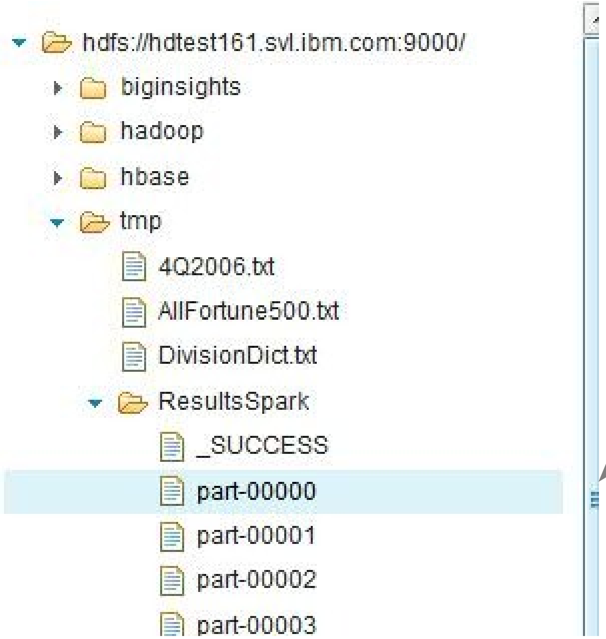
### Aggregated Metrics by Executor

| Executor ID | Address | Task Time | Total Tasks | Failed Tasks | Succeeded Tasks | Shuffle Read | Shuffle Write | Shuffle Spill (Memory) | Shuffle Spill (Disk) |
|---|---|---|---|---|---|---|---|---|---|
| 0 | hdtest161.svl.ibm.com:50145 | 17.1 m | 18 | 0 | 18 | 0.0 B | 0.0 B | 0.0 B | 0.0 B |
| 1 | hdtest162.svl.ibm.com:13943 | 7.8 m | 12 | 0 | 12 | 0.0 B | 0.0 B | 0.0 B | 0.0 B |

### Tasks

| Task Index | Task ID | Status | Locality Level | Executor | Launch Time | Duration | GC Time | Result Ser Time | Errors |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 1.5 m | 599 ms | 1 ms | |
| 1 | 1 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 28.2 s | 316 ms | | |
| 2 | 2 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 48.4 s | 521 ms | | |
| 3 | 3 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 48.1 s | 521 ms | | |
| 4 | 4 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 48.2 s | 521 ms | | |
| 5 | 5 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 47.1 s | 521 ms | | |
| 6 | 6 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 588 ms | | 1 ms | |
| 7 | 7 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 44.5 s | 521 ms | 1 ms | |
| 8 | 8 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 628 ms | | | |
| 9 | 9 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 48.3 s | 521 ms | | |
| 10 | 10 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 48.3 s | 521 ms | | |
| 11 | 11 | SUCCESS | PROCESS_LOCAL | hdtest162.svl.ibm.com | 2014/06/28 11:02:08 | 588 ms | | 1 ms | |
| 12 | 12 | RUNNING | PROCESS_LOCAL | hdtest161.svl.ibm.com | 2014/06/28 11:02:08 | 1.8 m | | | |

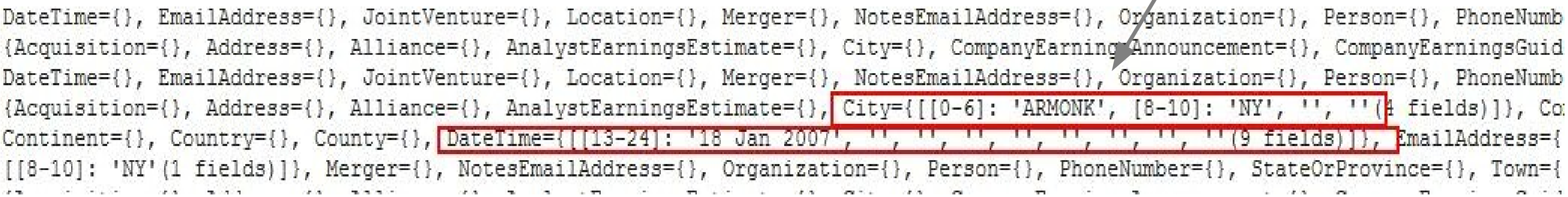
# Results from Named Entity Extraction (NEE) via Spark Java



- hdfs://hdtest161.svl.ibm.com:9000/
  - biginsights
  - hadoop
  - hbase
  - tmp
    - 4Q2006.txt
    - AllFortune500.txt
    - DivisionDict.txt
    - ResultsSpark
      - _SUCCESS
      - part-00000
      - part-00001
      - part-00002
      - part-00003

Results from NEE via Spark Java are persisted on IBM BigInsights' HDFS

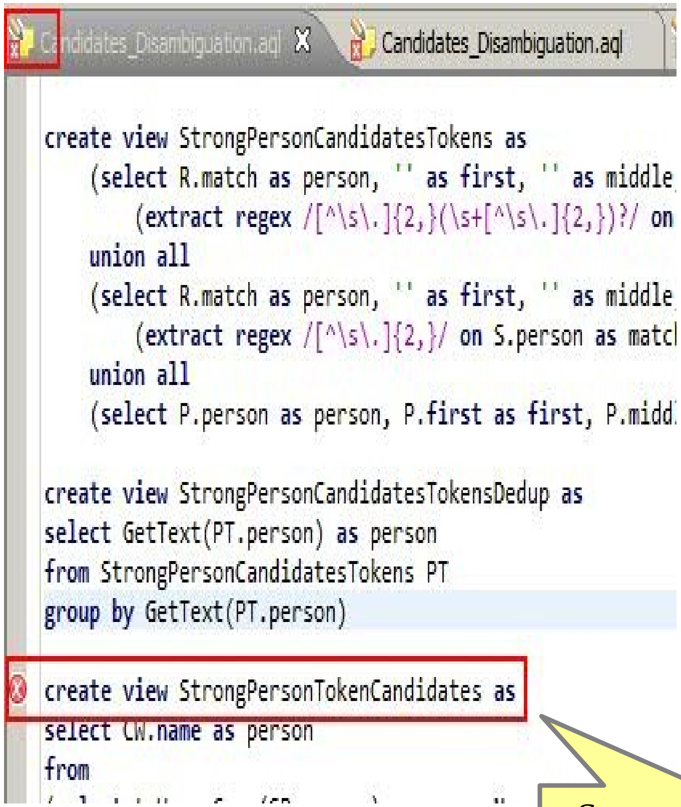
Results from NEE via Spark Java, per-document, hence sparse



```
DateTime={}, EmailAddress={}, JointVenture={}, Location={}, Merger={}, NotesEmailAddress={}, Organization={}, Person={}, PhoneNumb
{Acquisition={}, Address={}, Alliance={}, AnalystEarningsEstimate={}, City={}, CompanyEarningsAnnouncement={}, CompanyEarningsGuid
DateTime={}, EmailAddress={}, JointVenture={}, Location={}, Merger={}, NotesEmailAddress={}, Organization={}, Person={}, PhoneNumb
{Acquisition={}, Address={}, Alliance={}, AnalystEarningsEstimate={}, City={[[0-6]: 'ARMONK', [8-10]: 'NY', '', ''(4 fields)]}, Co
Continent={}, Country={}, County={}, DateTime={[[13-24]: '18 Jan 2007', '', '', '', '', '', '', '', ''(9 fields)]}, EmailAddress={
[[8-10]: 'NY'(1 fields)]}, Merger={}, NotesEmailAddress={}, Organization={}, Person={}, PhoneNumber={}, StateOrProvince={}, Town={
```

# Eclipse developer tool for text-analytics

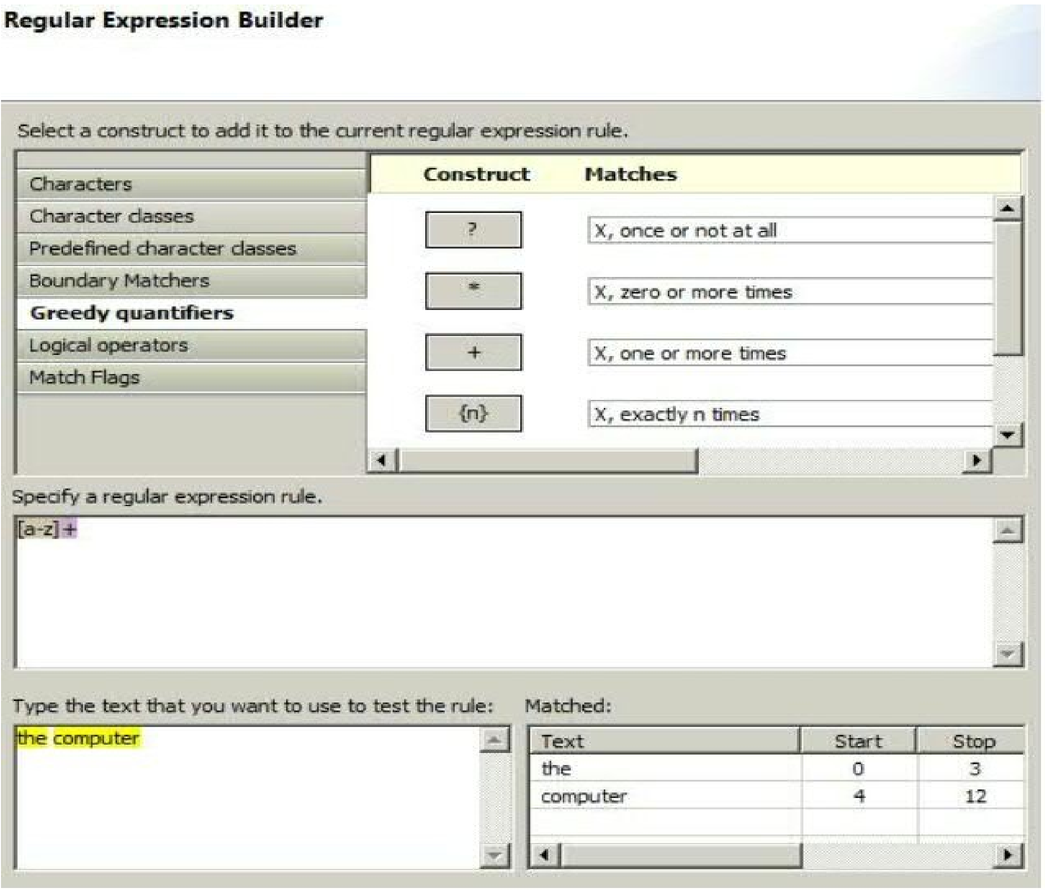
Build regular expressions with zero/little prior knowledge



```
create view StrongPersonCandidatesTokens as
    (select R.match as person, '' as first, '' as middle
        (extract regex /[^\s\.]{2,}(\s+[^\s\.]{2,})?/ on
    union all
    (select R.match as person, '' as first, '' as middle
        (extract regex /[^\s\.]{2,}/ on S.person as matc
    union all
    (select P.person as person, P.first as first, P.midd

create view StrongPersonCandidatesTokensDedup as
select GetText(PT.person) as person
from StrongPersonCandidatesTokens PT
group by GetText(PT.person)

create view StrongPersonTokenCandidates as
select CW.name as person
from
```



Select a construct to add it to the current regular expression rule.

| Construct | Matches |
|---|---|
| ? | X, once or not at all |
| * | X, zero or more times |
| + | X, one or more times |
| {n} | X, exactly n times |

Specify a regular expression rule.

[a-z]+

Type the text that you want to use to test the rule:

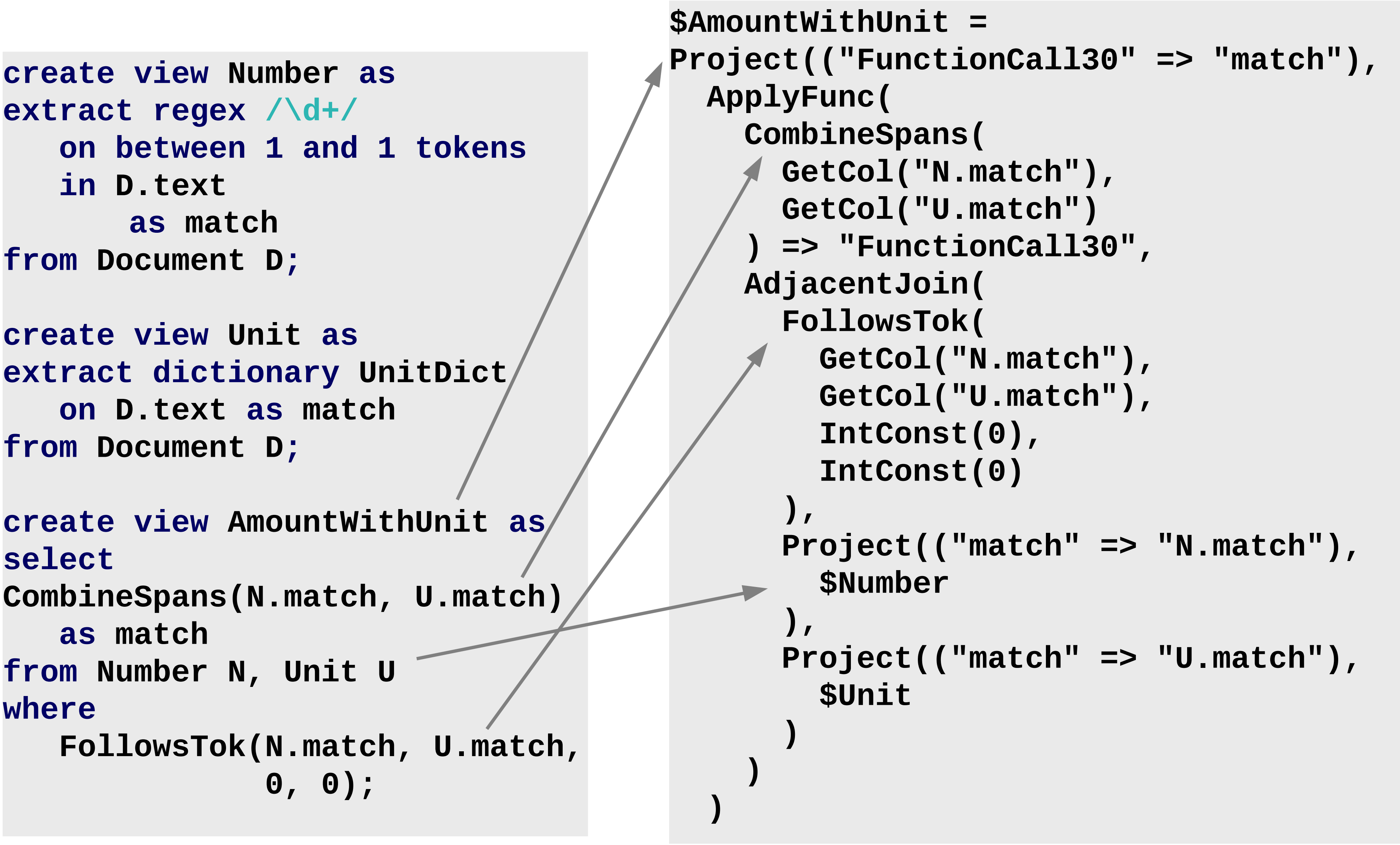
the computer

Matched:

| Text | Start | Stop |
|---|---|---|
| the | 0 | 3 |
| computer | 4 | 12 |

Syntax-highlighting, content-assist, markers etc.

# **AQL –** Source [*left*] → Sub-plan from compiled plan [*right*]



```
create view Number as
extract regex /\d+/
   on between 1 and 1 tokens
   in D.text
       as match
from Document D;

create view Unit as
extract dictionary UnitDict
   on D.text as match
from Document D;

create view AmountWithUnit as
select
CombineSpans(N.match, U.match)
   as match
from Number N, Unit U
where
   FollowsTok(N.match, U.match,
              0, 0);
```

```
$AmountWithUnit =
Project(("FunctionCall30" => "match"),
  ApplyFunc(
    CombineSpans(
      GetCol("N.match"),
      GetCol("U.match")
    ) => "FunctionCall30",
    AdjacentJoin(
      FollowsTok(
        GetCol("N.match"),
        GetCol("U.match"),
        IntConst(0),
        IntConst(0)
      ),
      Project(("match" => "N.match"),
        $Number
      ),
      Project(("match" => "U.match"),
        $Unit
      )
    )
  )
```