



# Business Intelligence via Natural Language Search

---

Designing a Query Engine for Capital Markets Data

November 2013 | Joel Bjornson | [joel.bjornson@dealogic.com](mailto:joel.bjornson@dealogic.com)



## Companies



Need Capital

## Investors



Have Capital

## Companies



Need Capital



Market Makers

## Investors



Have Capital

## Investment Banking

### ECM



Equity Capital Market

### DCM



Debt Capital Market

### Loan



### M&A



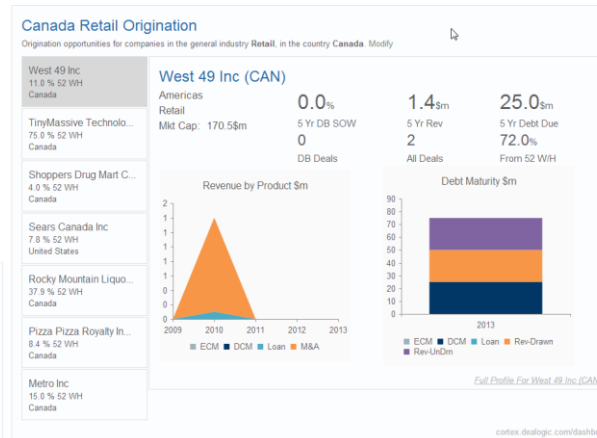
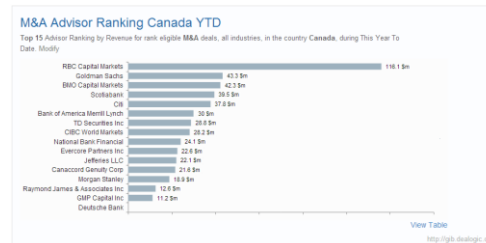
Merger & Acquisition

Deallogic has been tracking Capital Markets for more than 25 years



Investment Banks use Dealogic Analytics Products to:

- Analyze their performance
- Identify new business opportunities
- Keep track of their competitors



How to best serve data to our clients?

- Computer languages like SQL
- Pre-generated reports
- GUI for specifying search criteria and filters

How to best serve data to our clients?

- Computer languages like SQL
- Pre-generated reports
- GUI for specifying search criteria and filters
- English?





Signals

five biggest dcm healthcare countries by volume in europe compare last three years



Settings ▾

1 results in 0.29 seconds

[Rate This](#)

Showing results for: *Top 5 DCM Healthcare Country by Volume Europe 2010 vs 2011 vs 2012*

Search instead for:

[Top 5 DCM Healthcare-Products Country by Volume Europe 2010 vs 2011 vs 2012](#)

## DCM Countries by Volume Healthcare Europe

Top 5 Countries by **Volume** for rank eligible DCM deals, in the general industry **Healthcare**, in the sub-region **Europe**, comparing **2010**, **2011** and **2012**. [Modify](#)

#	Country	2010				2011				2012			
		#	Volume \$m ▾	Deals	Share %	#	Volume \$m	Deals	Share %	#	Volume \$m	Deals	Share %
1	Germany	1	7,291.9	7	46.2	2	2,155.8	3	16.0	3	2,929.8	4	12.9
2	Switzerland	2	5,307.9	2	33.6	6	398.5	2	3.0	2	4,958.1	3	21.9
3	Ireland	3	1,580.0	4	10.0	9	0.0	0	0.0	6	600.0	1	2.7
4	France	4	1,125.4	5	7.1	1	7,645.3	3	56.9	4	1,702.9	5	7.5
5	United Kingdom	5	381.2	1	2.4	3	1,269.4	2	9.4	1	10,182.2	6	45.0
<b>Subtotal</b>			<b>15,653.3</b>	<b>19</b>	<b>99.2</b>		<b>11,423.7</b>	<b>10</b>	<b>85.0</b>		<b>20,289.8</b>	<b>19</b>	<b>89.6</b>
<b>Total</b>			<b>15,787.1</b>	<b>20</b>	<b>100.0</b>		<b>13,446.0</b>	<b>15</b>	<b>100.0</b>		<b>22,632.6</b>	<b>28</b>	<b>100.0</b>

[View Chart](#)

Some challenges:

- English is ambiguous
- Must recognize millions of keywords
- Should tolerate spelling mistakes
- Needs to understand the semantics of queries
- Reports must be generated on the fly

Some challenges:

- English is ambiguous
- Must recognize millions of keywords
- Should tolerate spelling mistakes
- Needs to understand the semantics of queries
- Reports must be generated on the fly
- Has to be FAST!



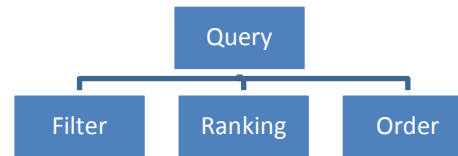
## English

high yield bond, high yield,  
jpm, jp morgan, goldman sachs,  
oil and gas, gas, oil, gasoline  
2010, last year, next week  
and, or, not, after, before  
walmart, ibm, american airlines  
uk, united kingdom, hungary,  
tech, computers, electronics  
...

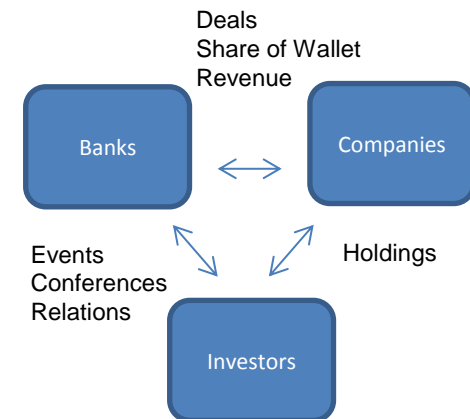
## Terms

- Category
- Industry
- Region
- DealType
- Product
- Bank
- Company
- Date
- Quantifier
- Unit
- Operator

## Semantic Query Model



## Reports



- Functional First .NET Language
- Roots in ML (OCam'l)
- Fully Supported in Visual Studio
- Multi Paradigm (Functional, Imperative, OO)
- Open Source

## Some example Terms:

```
type Date =  
  | Year of int  
  | Month of int  
  | YearMonth of int * int  
  | MonthDay of int * int  
  | Date of int * int * int
```

```
type FilterCriterion =  
  | RankEligible  
  | SponsorRelated  
  ...
```

```
type Product =  
  | ECM  
  | DCM  
  | Loan  
  | MNA
```

```
type Term =  
  | Date of Date  
  | FilterCriterion of FilterCriterion  
  | Product of Product  
  ...
```

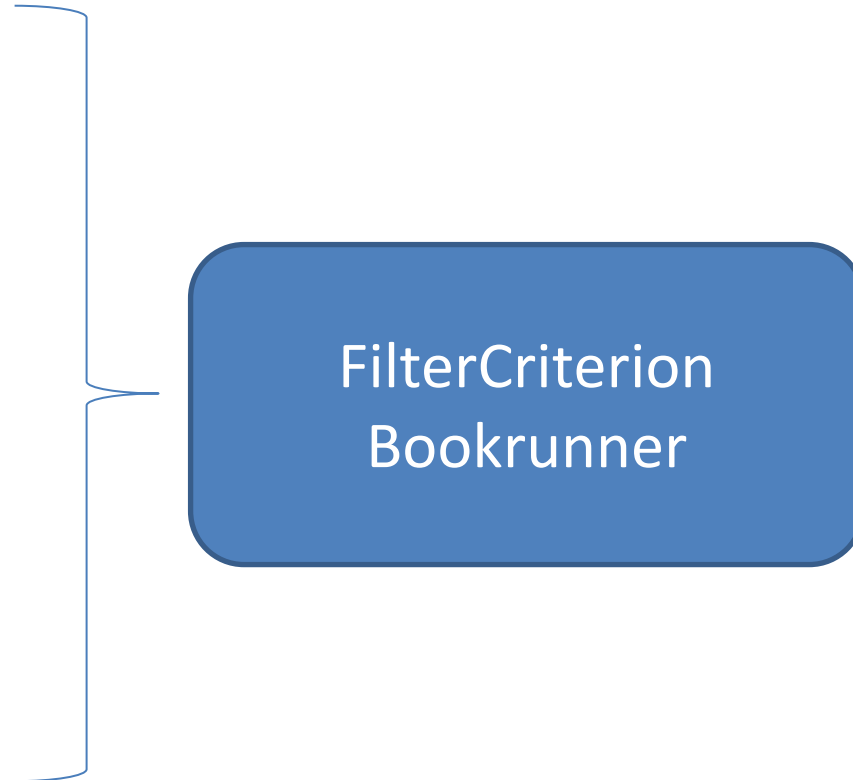
## Pattern Matching

```
let january = Term.Date (Date.Month 1)

/// Only matches terms representing January.
let matchJanuary term =
    match term with
    | Term.Date (Date.Month 1)
    | Term.YearMonth (_,1)
    | Term.MonthDay (1, _)      -> true
    | _                        -> false
```

Terms may have many synonyms

*bookrun  
bookrunner  
bookkeeper  
bookholder  
book-run  
book-runner  
book-keeper  
book-holder  
book run  
book runner  
book keeper  
book holder  
run by  
ran by  
running by  
keep by  
keeping by  
kept by  
book kept by  
...*





## Generating synonyms

```
let bookrunnerPattern =
  choose
    [
      // Ex: book runner
      !<[
        !["book"]
        !?["-"; " "]
        !["run"; "runner"; "keeper"; "holder"]
      ]
      // Ex: run by
      !<[
        !?["book"] ?> !?["-"; " "]
        !["run"; "ran"; "running"; "keep"; "keeping"; "kept"]
        ![" "]
        !["by"]
      ]
    ]

[FilterCriterion.Bookrunner, "Bookrunner", bookrunnerPattern ]
|> singleTermSearchItem FilterCriterion 1000.
```

*Norwegian Oil & Gas deals over 500\$m run by Deutsche Bank 2005-2010*

*Norwegian Oil & Gas deals over 500\$m run by Deutsche Bank 2005-2010*

[norwegian] [oil] [and] [gas] [deals] [over] [500] [\$] [m] [run] [by] [deutsche] [bank] [2005] [-] [2010]

## Norwegian Oil & Gas deals over 500\$m run by Deutsche Bank 2005-2010

[norwegian] [oil] [and] [gas] [deals] [over] [500] [\$] [m] [run] [by] [deutsche] [bank] [2005] [-] [2010]

Words	Terms	Score
<i>norwegian</i>	Region[Country[Nor, Norway]]	700
<i>oil and gas</i>	Industry[GIG[OILG, Oil & Gas]]	5000
<i>deals</i>	Category[Deals]	1000
<i>over</i>	Operator[GT]	250
<i>500</i>	Quantifier[500]	10000
<i>\$</i>	Currency[USD]	8500
<i>m</i>	Factor[Mega]	250
<i>run by</i>	FilterCriterion[Bookrunner]	1000
<i>deutsche bank</i>	Bank[7828379,DB]	3000
<i>2005</i>	Date[Year[2005]]	4000
<i>-</i>	Operator[Minus]	500
<i>2010</i>	Date[Year[2010]]	4000

- Assigning scores to the *importance* of synonyms and *Terms*
- Statistical methods – *Maximizing the likelihood of an interpretation based on search history and text analysis*
- Heuristic methods – Apply domain knowledge to improve the result

Signals

x
Q

Settings ▾

2 results in 0.08 seconds

Showing results for:  
Search instead for:  
[Ranking](#)

**Bank Ranki**

Top 10 Bank Ranking

#	Bank
1	JPMorgan
2	Bank of Ameri
3	Goldman Sach
4	Morgan Stanle
5	Citi
6	Deutsche Ban
7	Credit Suisse
8	Barclays
9	Wells Fargo S
10	UBS

<b>Subtotal</b>	30,021.4	11,231,999.7	7,628,162.9	12,365	55.2
<b>Total</b>	54,401.8	24,213,234.3	10,040,692.0	30,848	100.0

[View Chart](#)

**American Airlines Inc** Company Profile

The company **American Airlines Inc** (USA)

**Priced or Completed Transportation-Airlines Deals Americas** Priced or Completed Deals

Top 10 deals by Deal Value, **Transportation-Airlines, Americas**, that priced or completed Last Three Months.

**Priced or Completed American Airlines Inc Deals** Priced or Completed Deals

Top 10 deals by Deal Value **American Airlines Inc**, that priced or completed from 10/30/2003 to 10/30/2013.

**American Airlines Inc Deals Expected to Price or Complete October 2013** Deals Expected to Price or Complete

Top 10 deals by Deal Value **American Airlines Inc**, expected to price or complete October 2013.

**Bank Ranking American Airlines Inc** Bank Ranking

Top 10 Bank Ranking by Revenue, **American Airlines Inc**, from 10/30/2003 to 10/30/2013.

**Competitive Performance for American Airlines Inc** Competitive Performance

Competitive Performance by Revenue, **American Airlines Inc**, from 10/30/2003 to 10/30/2013.

**Industries by Revenue** Industry Ranking

Top 10 Industries by Revenue, YTD.

**Products by Revenue** Product Ranking

Products by Revenue, **American Airlines Inc**, in all industries, from 10/30/2003 to 10/30/2013.

<http://gib.dealogic.com>

**Signals**

×
Q

Settings ▾

2 results in 0.08 seconds

Showing results for:  
Search instead for:  
[Ranking](#)

### Bank Ranki

Top 10 Bank Ranking

#	Bank				
1	JPMorgan				
2	Bank of Ameri				
3	Goldman Sack				
4	Morgan Stanle				
5	Citi				
6	Deutsche Ban				
7	Credit Suisse				
8	Barclays	2,629.4	1,158,715.4	2,699,525.0	2,447 4.8
9	Wells Fargo Securities	1,618.4	569,485.3	1,407,541.0	2,284 3.0
10	UBS	1,592.8	537,611.6	1,305,421.8	1,421 2.9
	<b>Subtotal</b>	<b>30,021.4</b>	<b>11,231,999.7</b>	<b>7,628,162.9</b>	<b>12,365 55.2</b>
	<b>Total</b>	<b>54,401.8</b>	<b>24,213,234.3</b>	<b>10,040,692.0</b>	<b>30,848 100.0</b>

[View Chart](#)

**American Airlines Inc** *Company Profile*

The company **American Airlines Inc** (USA)

[Rate This](#)

---

**Priced or Completed American Airlines Inc Deals** *Priced or Completed Deals*

Top 10 deals by Deal Value **American Airlines Inc**, that priced or completed from 10/30/2003 to 10/30/2013.

---

**American Airlines Inc Deals Expected to Price or Complete October 2013** *Deals Expected to Price or Complete*

Top 10 deals by Deal Value **American Airlines Inc**, expected to price or complete October 2013.

---

**Bank Ranking American Airlines Inc** *Bank Ranking*

Top 10 Bank Ranking by Revenue, **American Airlines Inc**, from 10/30/2003 to 10/30/2013.

---

**Competitive Performance for American Airlines Inc** *Competitive Performance*

Competitive Performance by Revenue, **American Airlines Inc**, from 10/30/2003 to 10/30/2013.

---

**Industries by Revenue** *Industry Ranking*

Top 10 Industries by Revenue, YTD.

---

**Products by Revenue** *Product Ranking*

Products by Revenue, **American Airlines Inc**, in all industries, from 10/30/2003 to 10/30/2013.

<http://gib.dealogic.com>

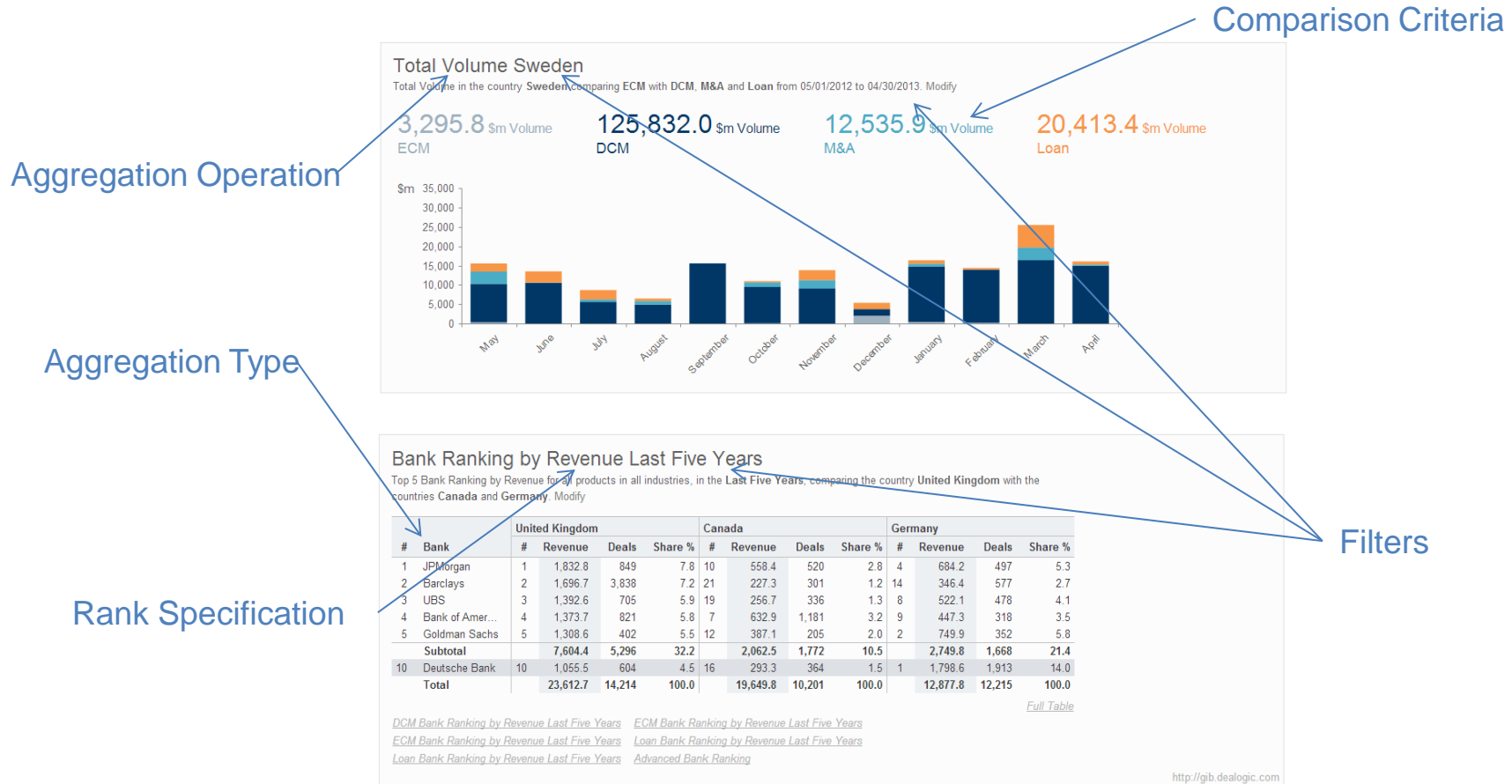
- A list of *Terms* does **not** capture meaning!
- How do we represent the *Semantics* of a query?



What is a Query?

- A set of *Filters* Ex: *Tech Europe*
- Potential Aggregation Operations Ex: *Total Volume*
- Rank specifications Ex: *Revenue*
- Aggregation Types Ex: *Product*
- Comparison Criteria Ex: *2010 vs 2011*

## Query Properties

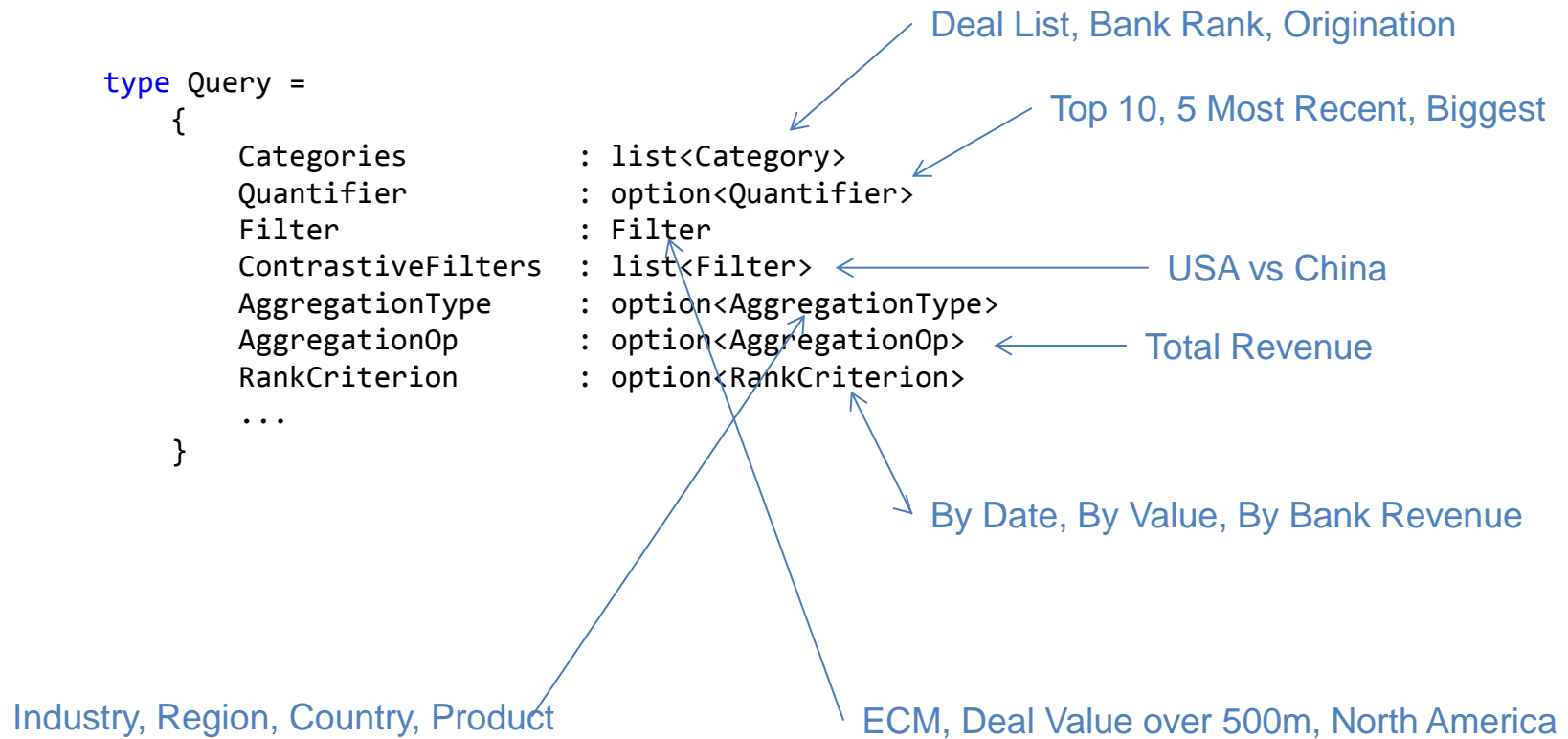


## Query Properties

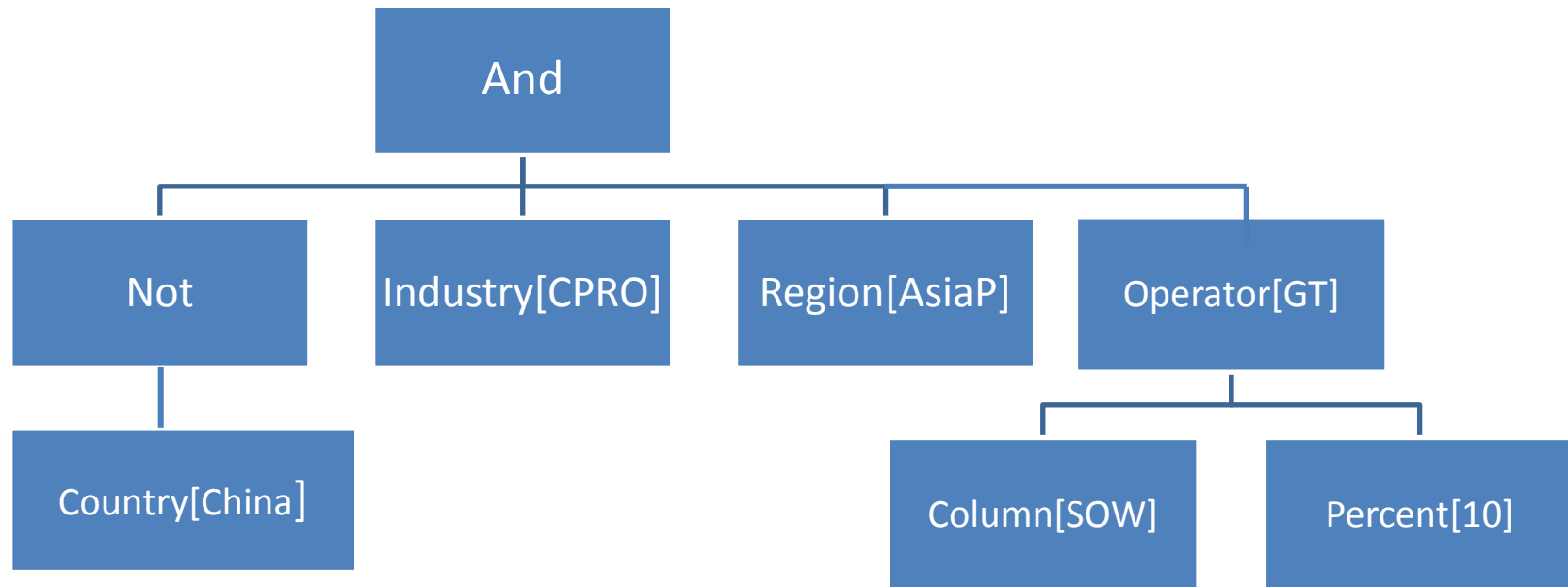
```

type Query =
{
  Categories
  Quantifier
  Filter
  ContrastiveFilters
  AggregationType
  AggregationOp
  RankCriterion
  ...
}

```



*Consumer Products Asiap except China for companies with sow over 10%*



## The Filter Type

```
type Filter =  
  | Empty  
  | Company of Company  
  | FinancialSponsor of FinancialSponsor  
  | Bank of Bank  
  | Product of Product  
  | DealType of DealType  
  | Region of Region  
  | Industry of Industry  
  | DateRange of DR.DateRange  
  | FilterCriterion of FilterCriterion  
  | RangeFilter of RangeFilter  
  | Not of Filter  
  | And of list<Filter>  
  | Or of list<Filter>
```

## Parsing Filters

```
// Ex: 2009 q3
```

```
Parse {  
  let! y = year  
  let! q = quarter  
  return DR.Quarter y q  
}
```

```
// Ex: Between 2007 and 2010
```

```
Parse {  
  do! between  
  let! d1 = date  
  do! maybe andOrTo  
  let! d2 = date  
  return DR.OfDates d1 d2 now  
}
```

```
// Ex: Two weeks before 2010-11-21.
```

```
Parse {  
  let! n = number  
  let! dru = dateRangeUnit  
  do! before  
  let! d = dateRange now  
  return beforeLastDateRange n d dru  
}
```

## Parsing Filters

```
// Ex: 2009 q3
Parse {
  let! y = year
  let! q = quarter
  return DR.Quarter y q
}

// Ex: Between 2007 and 2010
Parse {
  do! between
  let! d1 = date
  do! maybe andOrTo
  let! d2 = date
  return DR.OfDates d1 d2 now
}

// Ex: Two weeks before 2010-11-21.
Parse {
  let! n = number
  let! dru = dateRangeUnit
  do! before
  let! d = dateRange now
  return beforeLastDateRange n d dru
}
```

```
and filter1 now =
  parenFilter now
  <|> rangeFilter now
  <|> filterCriterion
  <|> notFilter now
  <|> dateRanges now
  <|> basicFilter now

and filter2 now =
  contrastiveFilters now
  <|> andFilter now
  <|> filter1 now

and filter3 now =
  orFilter now
  <|> (filter2 now)
```

## Normalizing Filters

- Constraints on applicable filters *Ex: Hungary or ECM*
- Filters interpreted literally are not always feasible *Ex: Hungary and Austria*
- Normalizations *ala* De Morgan
- Accommodate for imprecision *Ex: Not in Hungary or Austria*



## Normalizing Filters

```
let normalizeFilter =  
  rewrite <| function  
    | And []           -> Some Empty  
    | Or []            -> Some Empty  
    | Not Empty        -> Some Empty  
    | Not (Not e)      -> Some e  
    | Not (And xs)     -> List.map Not xs |> Or |> Some  
    | Not (Or xs)      -> List.map Not xs |> And |> Some  
    | And [x]          -> Some x  
    | Or [x]           -> Some x  
    | And xs           -> maybeRewrite And normalizeAnds xs  
    | Or xs            -> maybeRewrite Or normalizeOrs xs  
    | _                -> None
```

## Result Types:

- Deal Lists
- Ranking Reports
- Aftermarket Performance
- Aggregations
- Origination Tables
- Company Profiles
- Financial Sponsor Holdings

How should we match these queries?

*How much debt is due for coca cola next two years?*

*JP Morgan ECM over 100m*

*Best performing ECM banks in Brazil last year*

*Recent tech IPOs*

*How many deals over 500 m priced last year in Hungary?*

*What's pricing in Real Estate next month*

*Which were the top countries for tech deals last year*

*What's the share price of Facebook?*

## Rewrite Rules

- A *Rewrite Rule* is a pattern that matches and rewrites a list of *Query Expressions*
- Used to capture common query patterns and mappings to results
- Implemented using *Parsers*
- Can be applied recursively

## Rewrite Rules

```
// Ex: debt IBM
[
  keepInput debtMaturing
  keepInput EP.companyFilter
]
==> [aggOutstandingDebtRC]
|> withName "Debt maturing for company"
```

Debt due next year, maturing in two years, expiring..

IBM, Mol, American Airlines ...

```
// Ex: issued debt by IBM
[
  keepInput EP.byIssuedDebt
  keepInput EP.companyFilter
]
==> [aggIssuedDebtRC]
|> withName "Issued debt by company"
```

Accumulated debt, debt generated, issued debt

```
// Ex: debt issued
[
  keepInput (EP.outstanding <|> debt <|> EP.byOutstandingDebt)
  forget generated
]
==> [aggIssuedDebtRC]
|> withName "Debt generated"
```

**Signals**  × 🔍 Settings ▾

2 results in 0.23 seconds

[Rate This](#)

Showing results for: *Total by outstanding debt the company Coca-Cola Enterprises Inc Next Two... Years*

Search instead for:

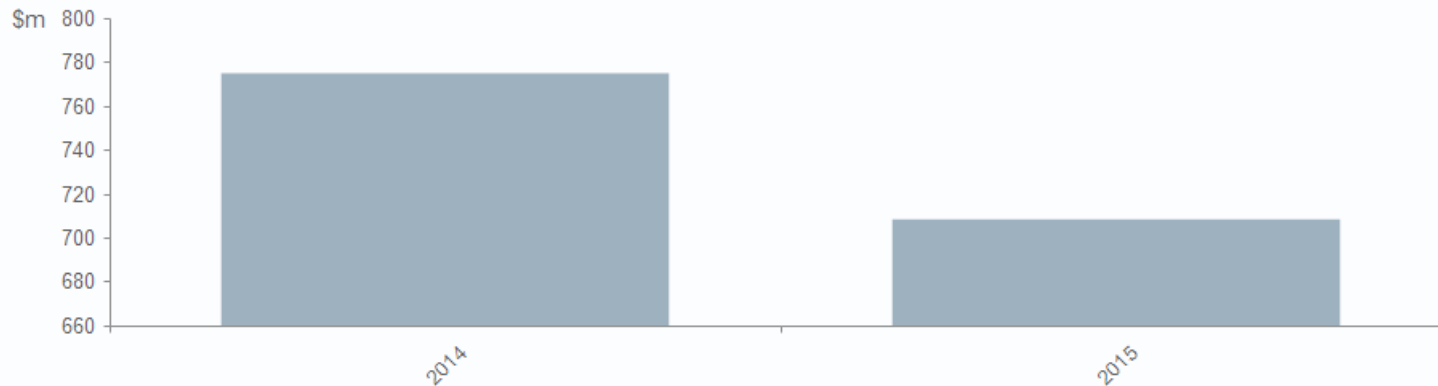
[Total by outstanding debt the company Coca-Cola Amatil Ltd Next Two Years](#)

[Total by outstanding debt the company Coca-Cola Next Two Years](#)

## Total Outstanding Debt - Coca-Cola Enterprises Inc

Total Volume of rank eligible deals **Maturing** in the **Next Two Years** for the company **Coca-Cola Enterprises Inc**. [Modify](#)

1,483.1 Debt \$m



## 1. Matching Words with Terms

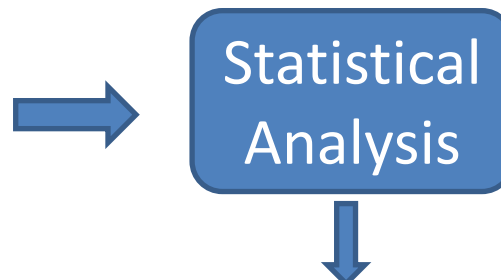
Score: 4600

SearchString	Terms	Priority	Error	Score	DisplayName
high yield	DealType[DealType[DCM-BOND-HY,Corporate Bond-High Yield]]	20000	0	20000	Corporate Bond-High Yield
bond	Product[DCM]	16000	0	16000	Debt Capital Market
energy	Industry[GIG[UTIL,Utility & Energy]]	5400	0	5400	Utility & Energy

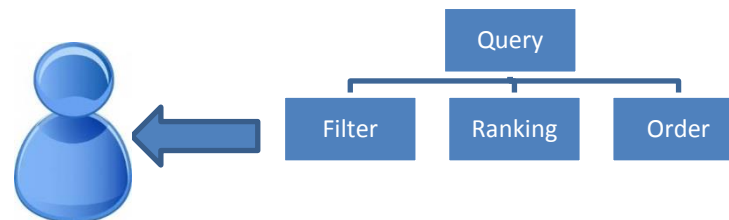
Score: 1405

SearchString	Terms	Priority	Error	Score	DisplayName
high	C[843489]	430	0	430	High Energy Batteries (India) Ltd
yield	C[648302]	650	0	650	Yield AB
bond	Product[DCM]	16000	0	16000	Debt Capital Market
energy	Industry[GIG[UTIL,Utility & Energy]]	5400	0	5400	Utility & Energy

## 2. Resolving Ambiguous Matches



## 3. Inferring Semantic Meaning



User Preferences

## 4. Matching Results

- Apply Rewrite rules
- Hand crafted rules
- Personal preferences
- Search history and user feed back (Not yet)

## 5. Generating Results

Priced or Completed Utility & Energy Deals in Europe

Top 5 Rank Eligible Corporate Bond High Yield deals for all products in the sub-region Europe, in the general industry Utility & Energy, that priced or completed within the Last Three Months. Middy

Company	Lead Bank	Value \$m	Type	Revenue \$m	Date
InterGen NV	Bank of America Merrill Lynch	800.0	DCM-BOND-HY	15.6	5/27/2013
OTEK Finance plc	Deutsche Bank	550.0	DCM-BOND-HY	1.3	4/23/2013
Lesenerge OAO	Sberbank CIB	95.0	DCM-BOND-HY	0.4	4/17/2013
OFEK Finance plc	Deutsche Bank	600.0	DCM-BOND-HY	5.6	3/27/2013
Nwen Finance plc	Barclays	268.7	DCM-BOND-HY	3.6	3/14/2013

Full Deal List

cortex.dealogic.com/dashboard

# Questions?