SystemT: an Algebraic Approach to Declarative Information Extraction

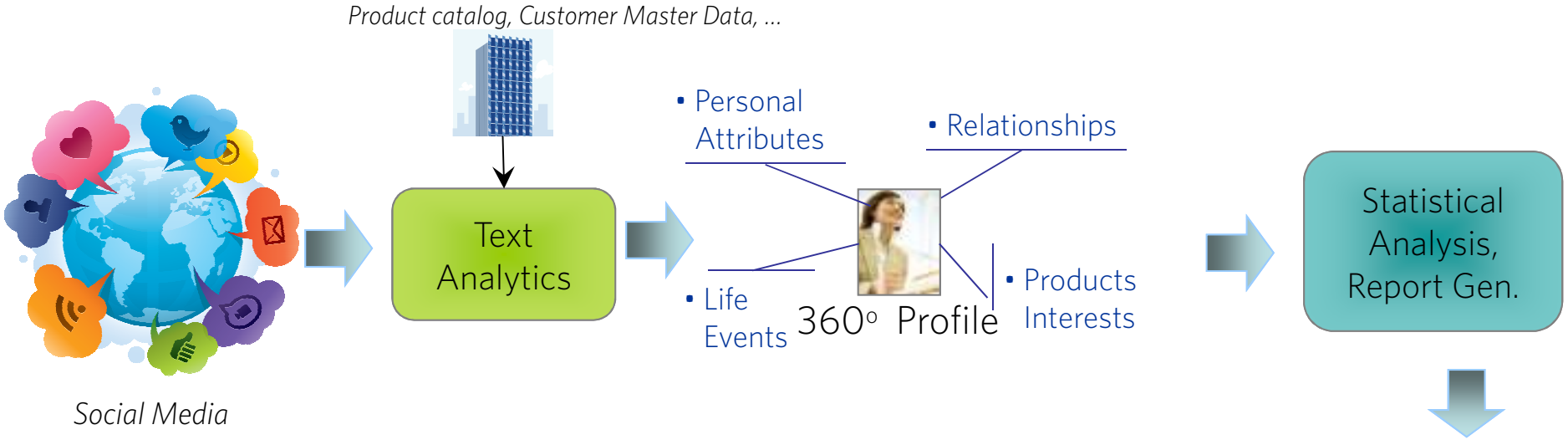
Yunyao Li

Scalable Natural Language Processing
IBM Research | Almaden

Outline

- Emerging Applications: Case Studies
- Enterprise Requirements
- SystemT : an Algebraic approach
  - Support Enterprise Information
  - Multilingual Support
  - Machine Learning
  - Tooling
- SystemT Courses
- Summary

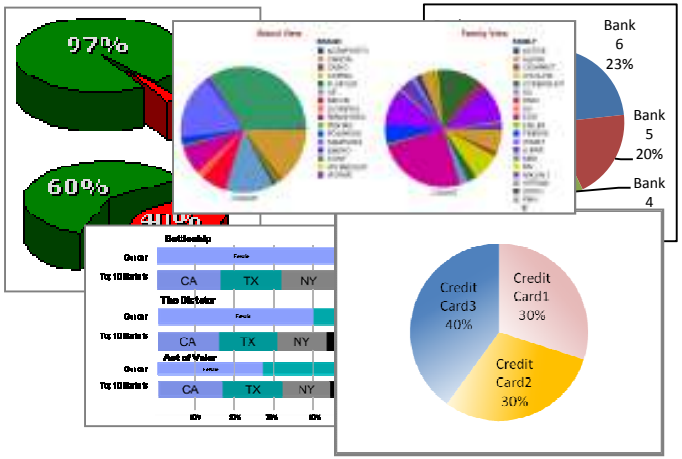
Case Study 1: Sentiment Analysis



Complexity
Sarcasm, wishful thinking, ...

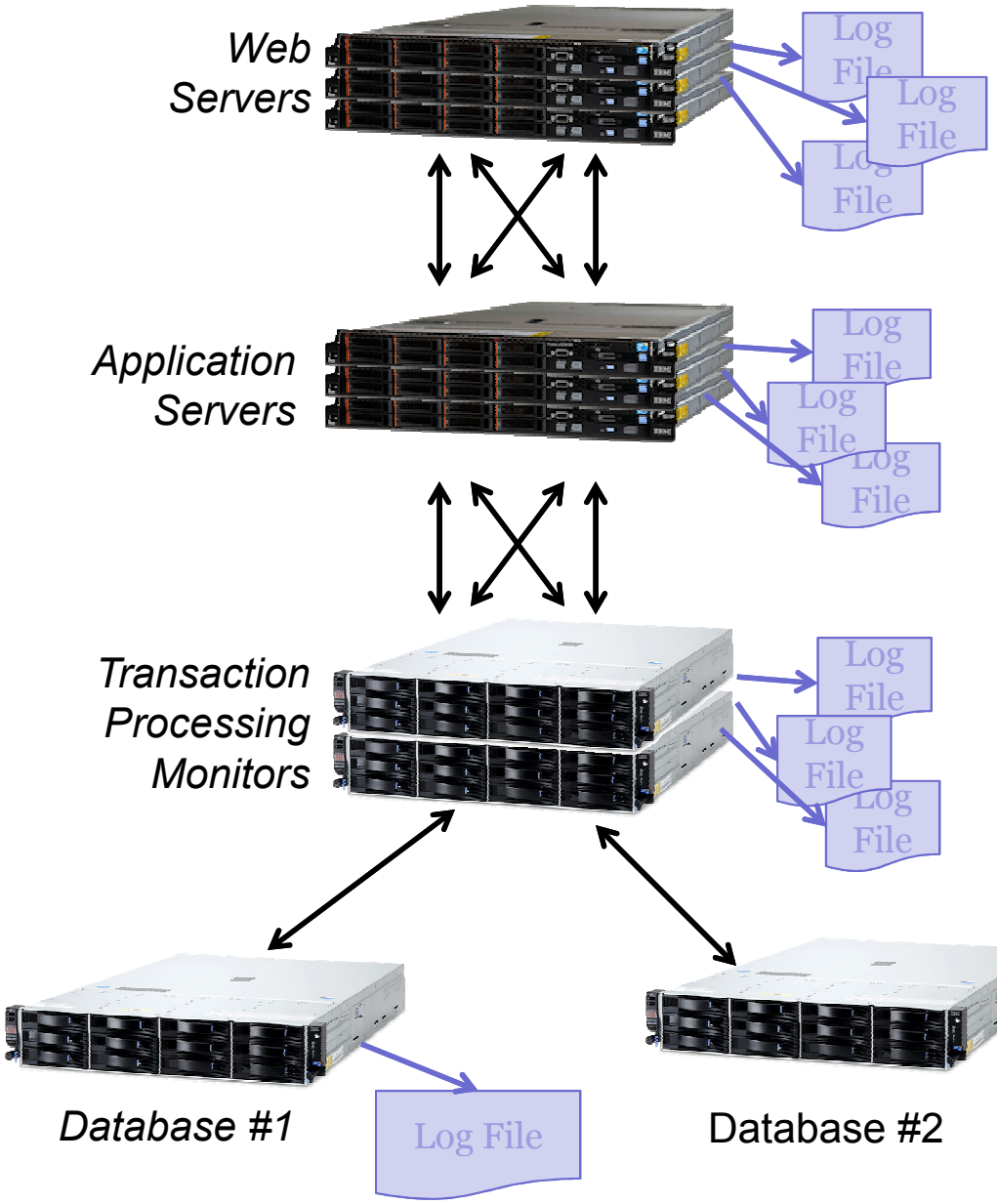
Breadth
Buzz, intent, sentiment, life events, personal atts, ...

Scale
500M+ tweets a day, 100M+ consumers, ...



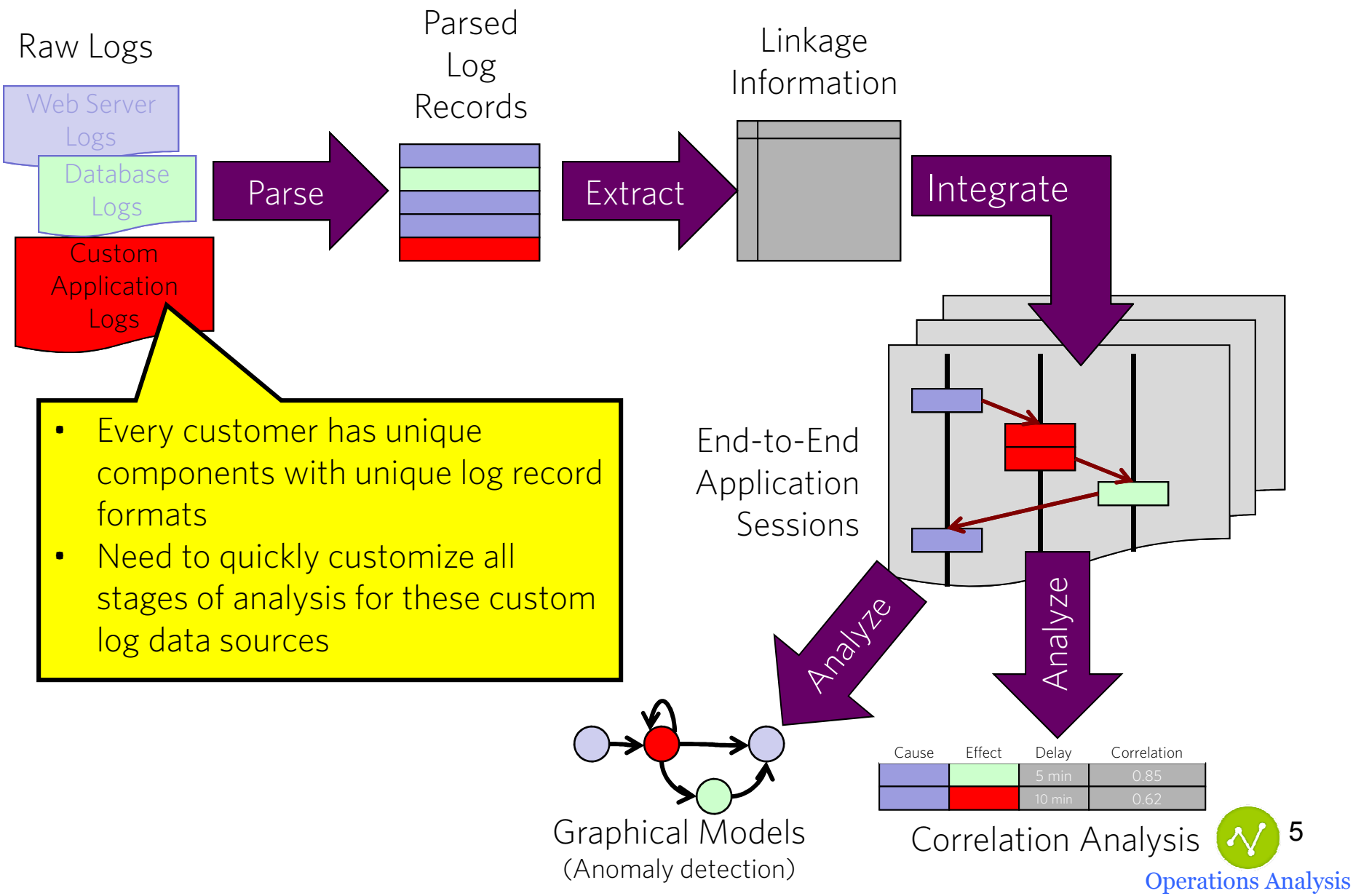
Customer 360º

Case Study 2: Machine Analysis



- Web site with multi-tier architecture
- Every component produces its own system logs
- An error shows up in the log for Database #2
- What sequence of events led to this error?

Case Study 2: Machine Analysis



- Every customer has unique components with unique log record formats
- Need to quickly customize all stages of analysis for these custom log data sources

Outline

- Emerging Applications: Case Studies
- Enterprise Requirements
- SystemT : an Algebraic approach
  - Support Enterprise Information
  - Multilingual Support
  - Machine Learning
  - Tooling
- SystemT Courses
- Summary

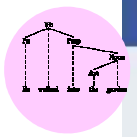
Text Analytics vs Information Extraction

POS Tagging

… NNS JJ IN NNP IN NNP …

Text Normalization

I am going to go see Gozilla in 3 months.



Dependency Parsing

- Applications
- Clustering
- Classification
- Regression

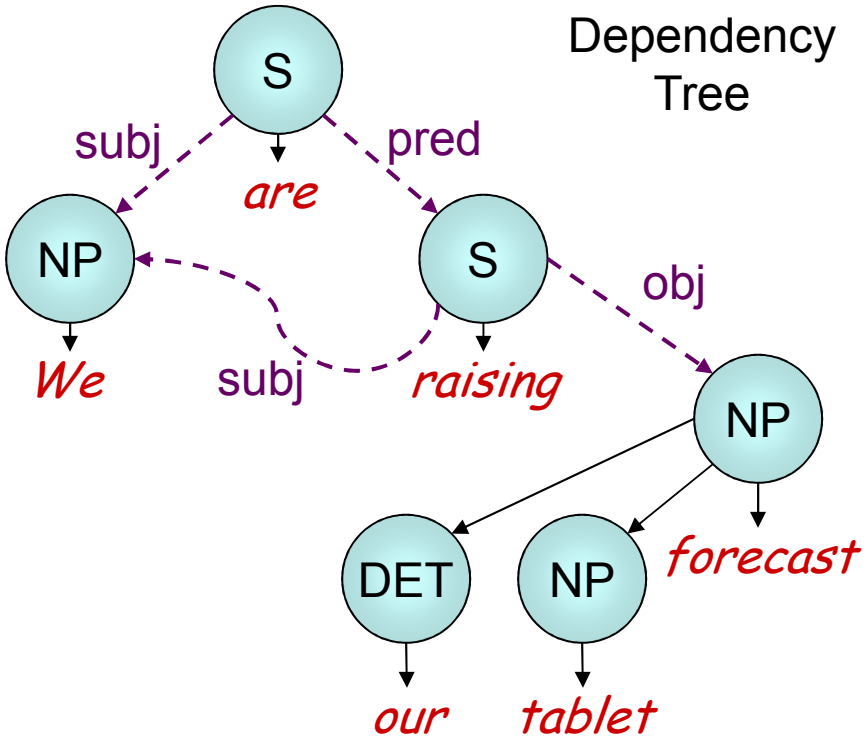
Enterprise Requirements

- **Expressivity**
  - Need to express complex NLP algorithms for a variety of tasks and data sources
- **Scalability**
  - Large data volumes, often orders of magnitude larger than classical NLP corpora
    - Social Media: Twitter alone has 500M+ messages / day; 1TB+ per day
    - Financial Data: SEC alone has 20M+ filings, several TBs of data, with documents range from few KBs to few MBs
    - Machine Data: One application server under moderate load at medium logging level → 1GB of logs per day
- **Transparency**
  - Every customer's data and problems are unique in some way
  - Need to easily comprehend, debug and enhance extractors

Expressivity Example: Different Kinds of Parses

Natural Language

We are raising our tablet forecast.



Machine Log

Oct 1 04:12:24 9.1.1.3 41865: %PLATFORM_ENV-1-DUAL_PWR: Faulty internal power supply B detected

Time	*Oct 1 04:12:24*
Host	*9.1.1.3*
Process	*41865*
Category	*%PLATFORM_ENV-1-DUAL_PWR*
Message	*Faulty internal power supply B detected*

Expressivity Example: Fact Extraction (Tables)

PUBLIC UTILITIES BOARD AND ITS SUBSIDIARIES

STATEMENTS OF COMPREHENSIVE INCOME

Year ended 31 March 2012

	Note	GROUP 31 March 2012 S$'000	GROUP 31 March 2011 S$'000	BOARD 31 March 2012 S$'000	BOARD 31 March 2011 S$'000
Operating income	3	1,037,549	1,010,737	1,030,740	1,005,426
Operating expenses	4	(1,037,056)	(998,773)	(1,030,647)	(993,502)
Net operating income		493	11,964	93	11,924
Non-operating income	5	36,000	19,768	37,881	19,771
Net income before financing expenses and operating grants		36,493	31,732	37,974	31,695
Financing expenses	6	(108,030)	(103,608)	(108,030)	(103,608)
Net Loss before operating grants		(71,537)	(71,876)	(70,056)	(71,913)
Operating grants from government	13.1	199,035	185,218	199,035	185,218
Net income after grants and before contribution to government consolidated fund and taxation		127,498	113,342	128,979	113,305
Contribution to government consolidated fund and taxation	7	(20,230)	(19,269)	(20,231)	(19,369)
Net income after grants and after contribution to government consolidated fund and taxation		97,268	94,073	98,748	94,044
Other comprehensive income		–	–	–	–
Total comprehensive income for the year		97,268	94,073	98,748	94,044
Attributable to:					
Shareholder of the Board	20.3	97,268	94,073	98,748	94,044

Identify line item for Operating expenses from Income statement (financial table in pdf document)

Singapore 2012 Annual Report
(136 pages PDF)

Identify note breaking down Operating expenses line item, and extract opex components

4 OPERATING EXPENSES

	Note	GROUP 31 March 2012 S$'000	GROUP 31 March 2011 S$'000	BOARD 31 March 2012 S$'000	BOARD 31 March 2011 S$'000
Direct operating expenses					
- electricity		147,427	126,539	147,427	126,539
- manpower		177,901	185,272	177,852	185,128
- depreciation		264,431	254,436	264,431	253,753
- plant rental		10,071	24,801	10,071	24,801
- property tax		15,014	14,365	15,014	14,365
- maintenance and others	4.1	293,002	266,880	286,642	262,436
Indirect operating expenses					
- service departments' costs	4.2	129,210	126,480	129,210	126,480
	4.3	1,037,056	998,773	1,030,647	993,502

Expressivity Example: Sentiment Analysis

Analyst Research Reports

- *Intel's 2013 capex is elevated at 23% of sales, above average of 16%*
- *IBM announced 4Q2012 earnings of $5.13 per share, compared with 4Q2011 earnings of $4.62 per share, an increase of 11 percent*
- *We continue to rate shares of MSFT neutral.*
- *FHLMC reported $4.4bn net loss and requested $6bn in capital from Treasury.*
- *Sell EUR/CHF at market for a decline to 1.31000…*

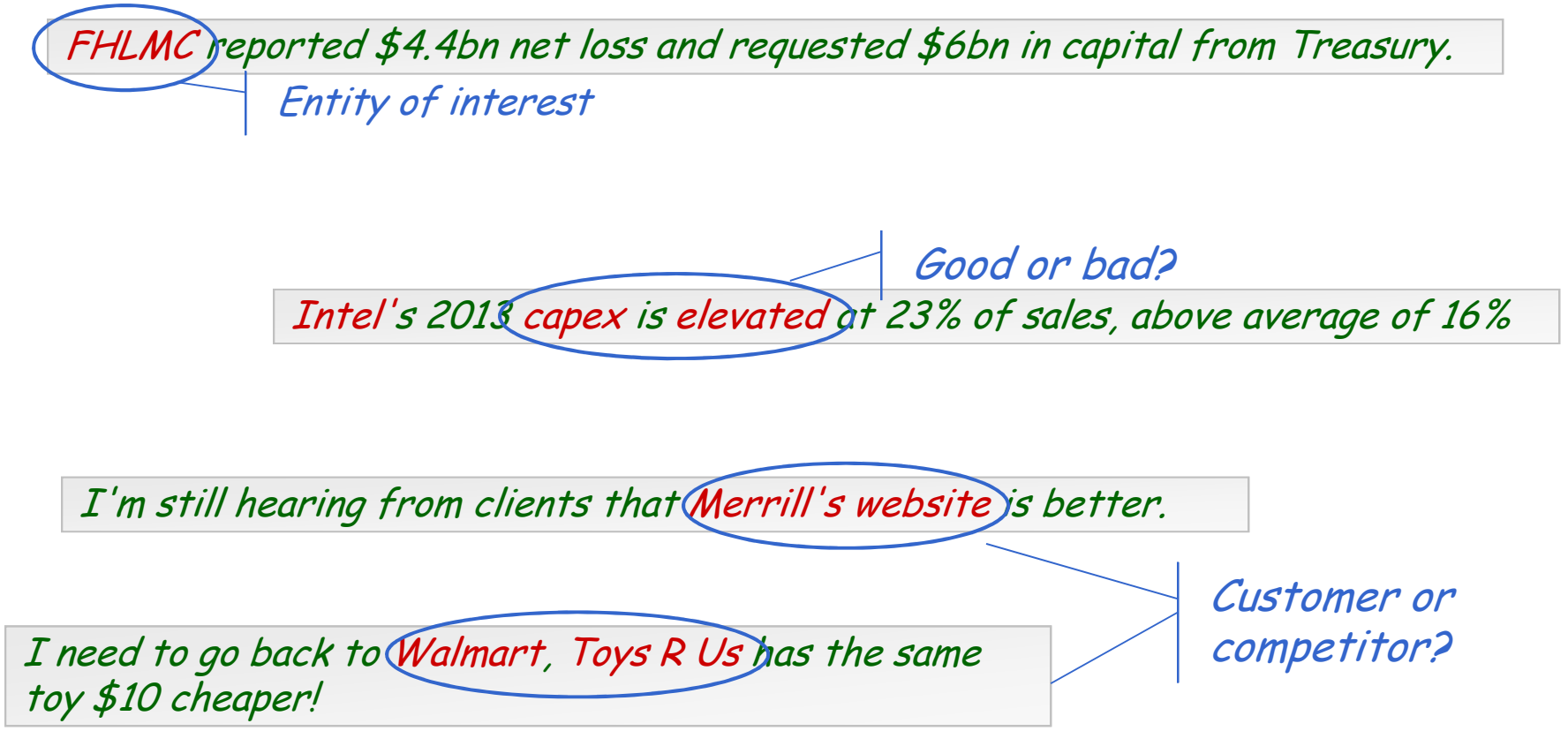
Customer Surveys

- *Not a pleasant client experience. Please fix ASAP.*
- *I'm still hearing from clients that Company A's website is better.*
- *X… fixing something that wasn't broken*

Social Media

- *Makin chicken fries at home bc everyone sucks!*
- *Bank X got me ****ed up today!*
- *Mcdonalds mcnuggets are fake as shit but they so delicious.*
- *You are never too old for Disney movies.*
- *We should do something cool like go to Z (**kidding**).*

Transparency Example: Need to incorporate in-situ data

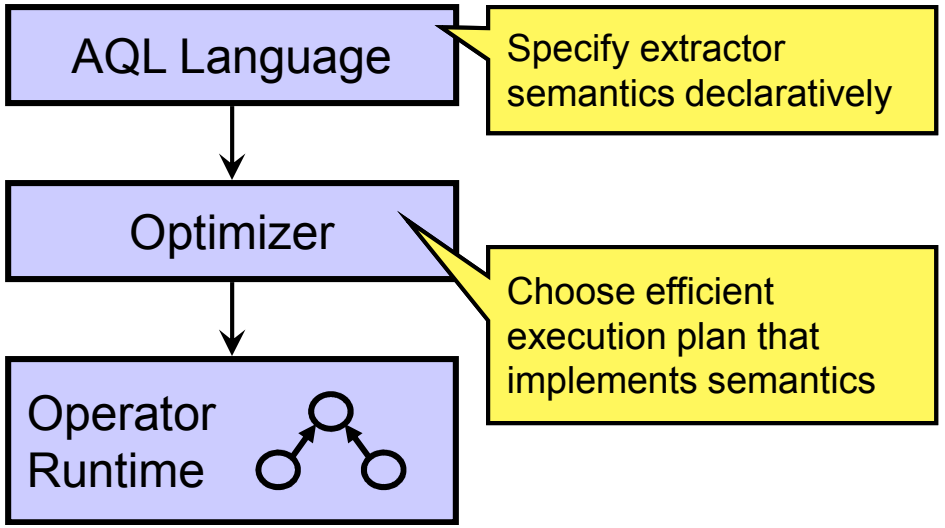


Outline

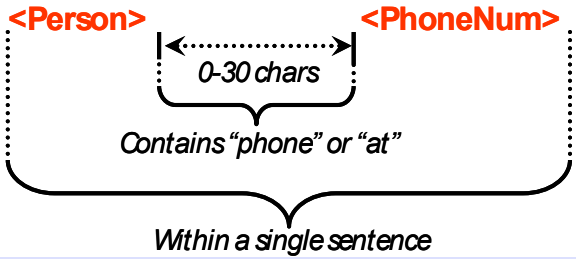
- Emerging Applications: Case Studies
- Enterprise Requirements
- SystemT : an Algebraic approach
  - Support Enterprise Information
  - Multilingual Support
  - Machine Learning
  - Tooling
- SystemT Courses
- Summary

SystemT: IBM Research project started in 2006

SystemT Architecture



Example AQL Extractor

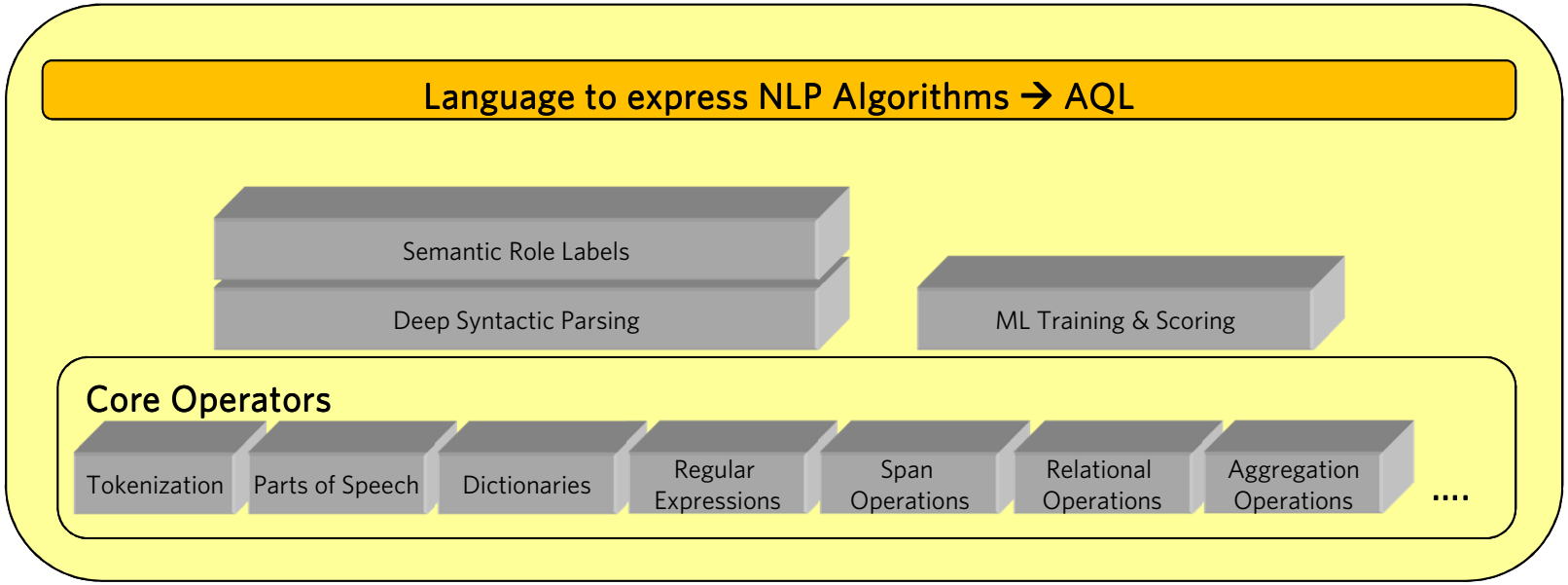



```
create view PersonPhone as
select P.name as person, N.number as phone
from Person P, PhoneNumber N, Sentence S
where
    Follows(P.name, N.number, 0, 30)
    and Contains(S.sentence, P.name)
    and Contains(S.sentence, N.number)
    and ContainsRegex(/\b(phone|at)\b/,
        SpanBetween(P.name, N.number));
```

Fundamental Results & Theorems

- Expressivity: The class of extraction tasks expressible in AQL is a strict superset of that expressible through cascaded regular automata.
- Performance: For any acyclic token-based finite state transducer *T*, there exists an operator graph *G* such that evaluating *T* and *G* has the same computational complexity.

Expressivity: Rich Set of Operators



Expressivity: AQL vs. Custom Java Code

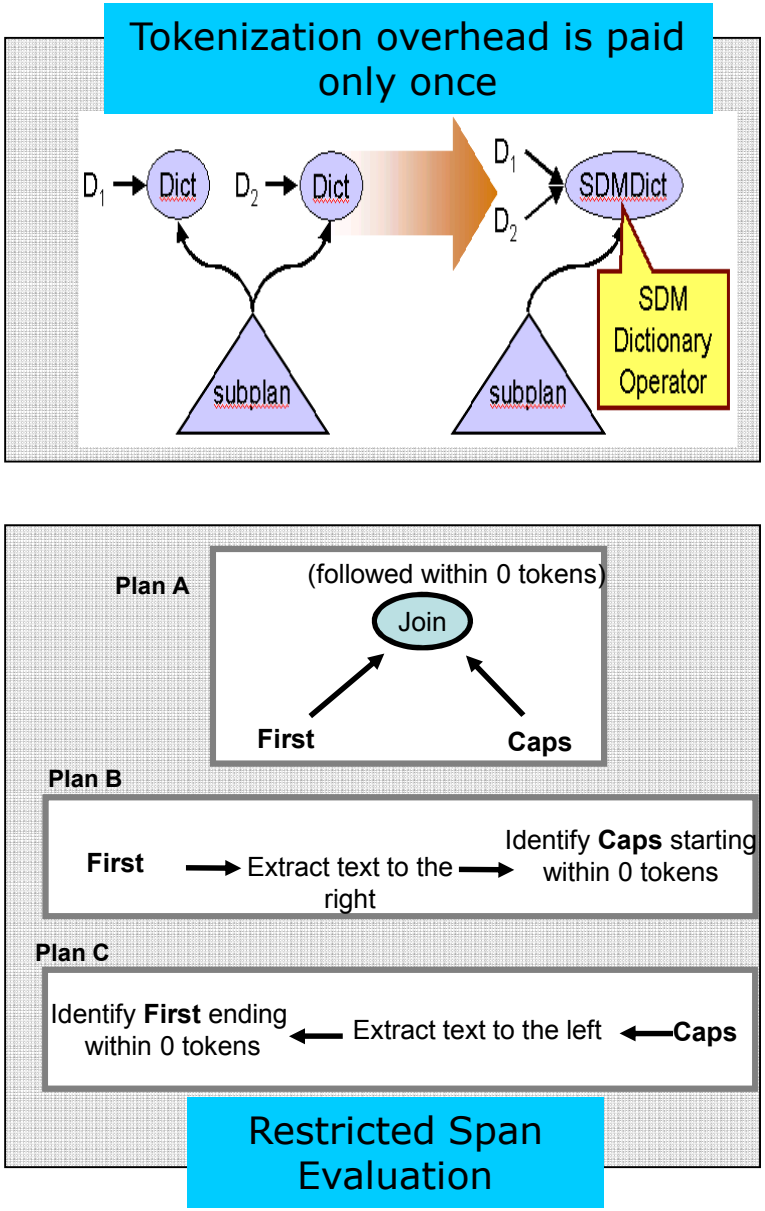
```
create dictionary AbbrevDict from file
    'abbreviation.dict';

create view SentenceBoundary as
select R.match as boundary
from ( extract regex /(([\.\?!]+\s)|(\n\s*\n))/
            on D.text as match from Document D ) R
where
    Not(ContainsDict('AbbrevDict',
        CombineSpans(LeftContextTok(R.match, 1),R.match)));
```

Equivalent AQL Implementation

Java Implementation of Sentence Boundary Detection

Scalability: Class of Optimizations in SystemT

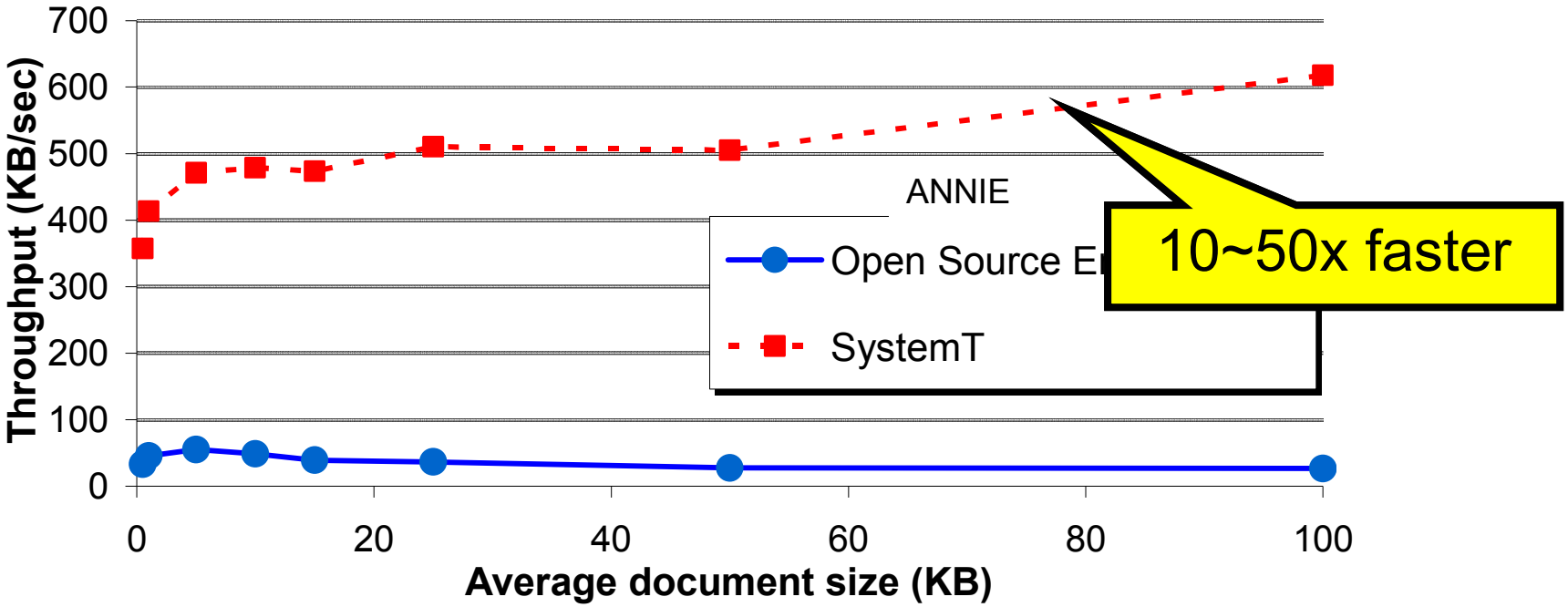
- **Rewrite-based**: rewrite algebraic operator graph
  - Shared Dictionary Matching
  - Shared Regular Expression Evaluation
  - On-demand tokenization
- **Cost-based**: relies on novel selectivity estimation for text-specific operators
  - Standard transformations
    - E.g., push down selections
  - Restricted Span Evaluation
    - Evaluate expensive operators on restricted regions of the document



Performance Comparison (with ANNIE)

Task: Named Entity

Dataset : Different document collections from the Enron corpus obtained by randomly sampling 1000 documents for each size



[Chiticariu et al., ACL'10]

Performance Comparison on Larger Documents

Datasets : Web crawl and filings from the Securities and Exchanges Commission (SEC)

Dataset	*Document Size*		*Throughput (KB/sec)*		*Average Memory (MB)*	
	Range	Average	ANNIE	SystemT	ANNIE	SystemT
Web Crawl	68 B – 388 KB	8.8 KB	42.8	498.8	201.8	77.2
Mec SEC I						
La SEC						

Theorem: For any acyclic token-based FST *T*, there exists an operator graph *G* such that evaluating *T* and *G* has the same computational complexity

Throughput benefits carryover fo wide-variety of document sizes

Much lower memory footprint

AQL: Going beyond Feature Extraction

Extraction Task: Named-entity extraction
Systems compared: SystemT (customized) vs. [Florian et al.'03] [Minkov et al.'05]

Dataset	Entity Type	System	Precision	Recall	F-measure
CoNLL 2003	Location	SystemT	93.11	91.61	92.35
		Florian	90.59	91.73	91.15
	Organization	SystemT	92.25	85.31	88.65
		Florian	85.93	83.44	84.67
	Person	SystemT	96.32	92.39	94.32
		Florian	92.49	95.24	93.85
Enron	Person	SystemT	87.27	81.82	84.46
		Minkov	81.1	74.9	77.9

Transparency without machine learning outperforms machine learning without transparency.

[Chiticariu et al., EMNLP'10]

Outline

- Emerging Applications: Case Studies
- Enterprise Requirements
- SystemT : an Algebraic approach
  - Support Enterprise Information
  - Multilingual Support
  - Machine Learning
  - Tooling
- SystemT Courses
- Summary

Multilingual Support

Tokenization	• 26+ languages • All other languages supported via whitespace/punctuation tokenizer
Part of Speech and Lemmatization	• 18+ languages, including English and Western languages, Arabic, Russian, CJK
Semantic Role Labeling	• 6+ major languages • Ongoing work on Multilingual SRL [Akbik et al, ACL'15]
Annotator Libraries	• Out-of-the-box customizable libraries for multiple applications and data sources in multiple languages

Outline

- Emerging Applications: Case Studies
- Enterprise Requirements
- SystemT : an Algebraic approach
  - Support Enterprise Information
  - Multilingual Support
  - Machine Learning
  - Tooling
- SystemT Courses
- Summary

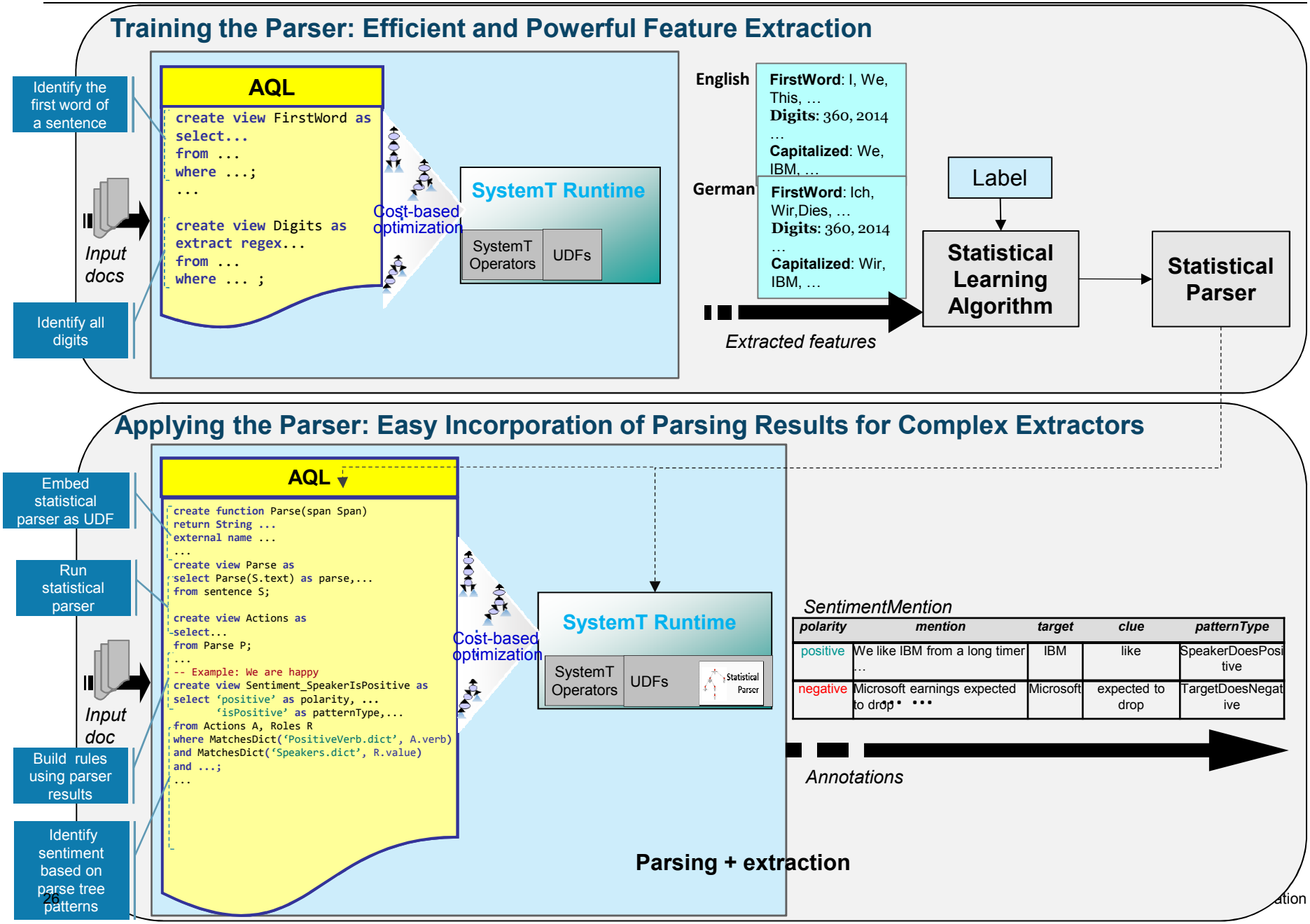
Machine Learning in SystemT

- AQL provides a foundation of transparency
- Next step: Add machine learning without losing transparency
- Major machine learning efforts:
  - Embeddable Models in AQL
  - Learning using AQL as target language

Machine Learning in SystemT

- AQL provides a foundation of transparency
- Next step: Add machine learning without losing transparency
- Major machine learning efforts:
  - Embeddable Models in AQL
    - Model Training and Scoring → integration with SystemML
    - Deep Parsing & Semantic Role Labeling [Akbik et al., ACL'15]
    - Text Normalization [Zhang et al. ACL'13, Li & Baldwin, NAACL '15]
  - Learning using AQL as target language

Simplify Training and Applying Statistical Parsers



SystemML in a Nutshell

- Provides a language for data scientists to implement machine learning algorithms
  - Declarative, high-level language with R-like syntax (also Python)
  - Also comes with approx. 20 algorithms pre-implemented

- Compiles execution plans ranging from single node (scale up multi threaded) to scale out (MapReduce, Spark)
  - Cost-based optimizer to generate execution plans, parallelize
    - Based on data and system characteristics
  - Operators for in-memory single node and cluster execution

- Runs in embeddable, standalone, and cluster mode
- Supports various APIs

- Apache SystemML Incubator project: http://systemml.apache.org
- Ongoing research effort at IBM Research - Almaden

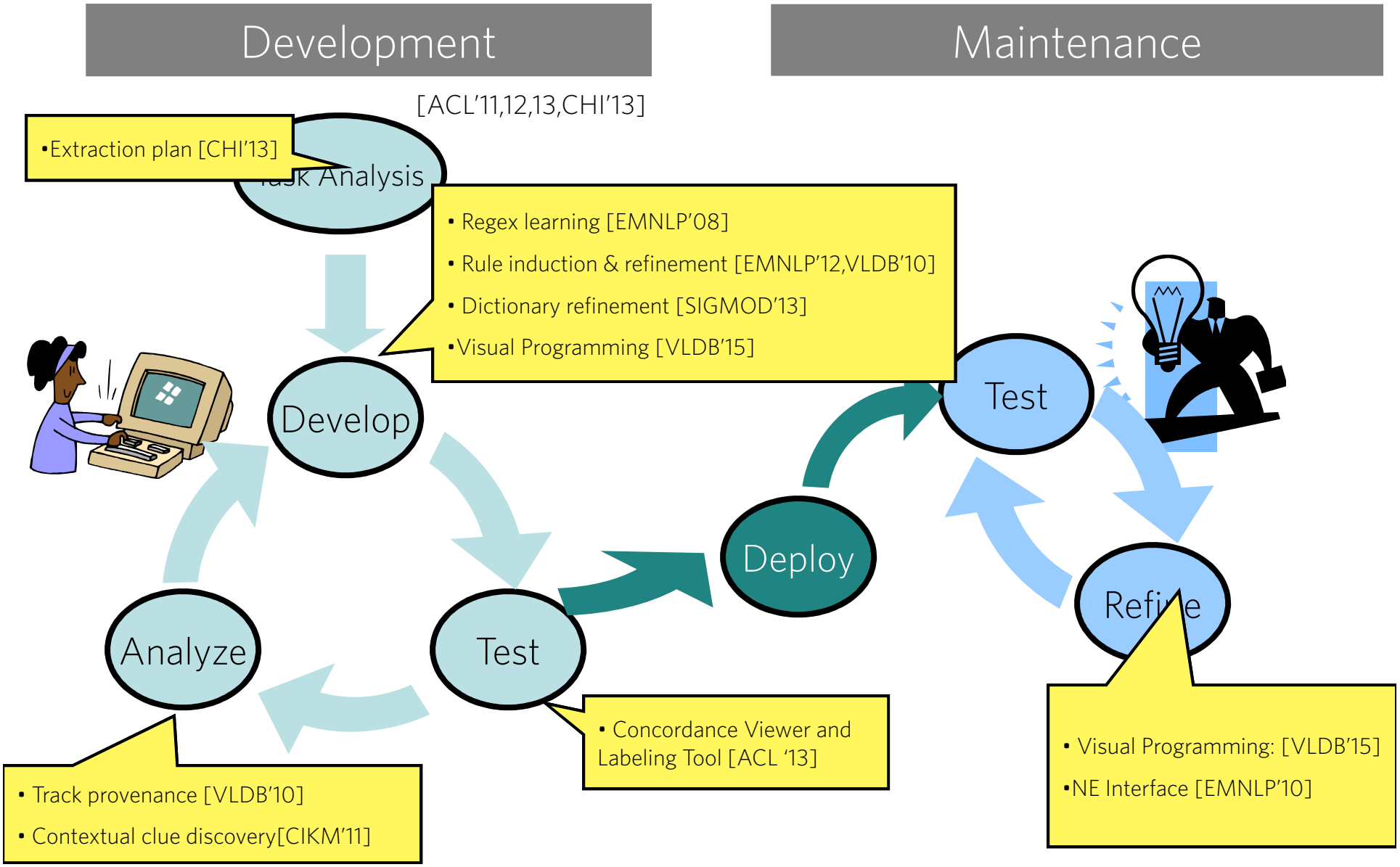
Machine Learning in SystemT

- AQL provides a foundation of transparency
- Next step: Add machine learning without losing transparency
- Major machine learning efforts:
  - Embeddable Models in AQL
  - Learning using AQL as target language
    - Low-level features: regular expressions [Li et al., EMNLP'08], dictionaries [Li et al., CIKM'11, Roy et al., SIGMOD'13]
    - Rule refinement [Liu et al., VLDB'10]
    - Rule induction [Nagesh et al., EMNLP'12]
    - Ongoing research

Outline

- Emerging Applications: Case Studies
- Enterprise Requirements
- SystemT : an Algebraic approach
  - Support Enterprise Information
  - Multilingual Support
  - Machine Learning
  - Tooling
- SystemT Courses
- Summary

Tooling Research for Productivity



Eclipse Tools Overview

Ease of Programming

AQL Editor: syntax highlighting, auto-complete, hyperlink navigation

Result Viewer: visualize/compare/evaluate

Explain: show how each result was generated

Workflow UI: end-to-end development wizard

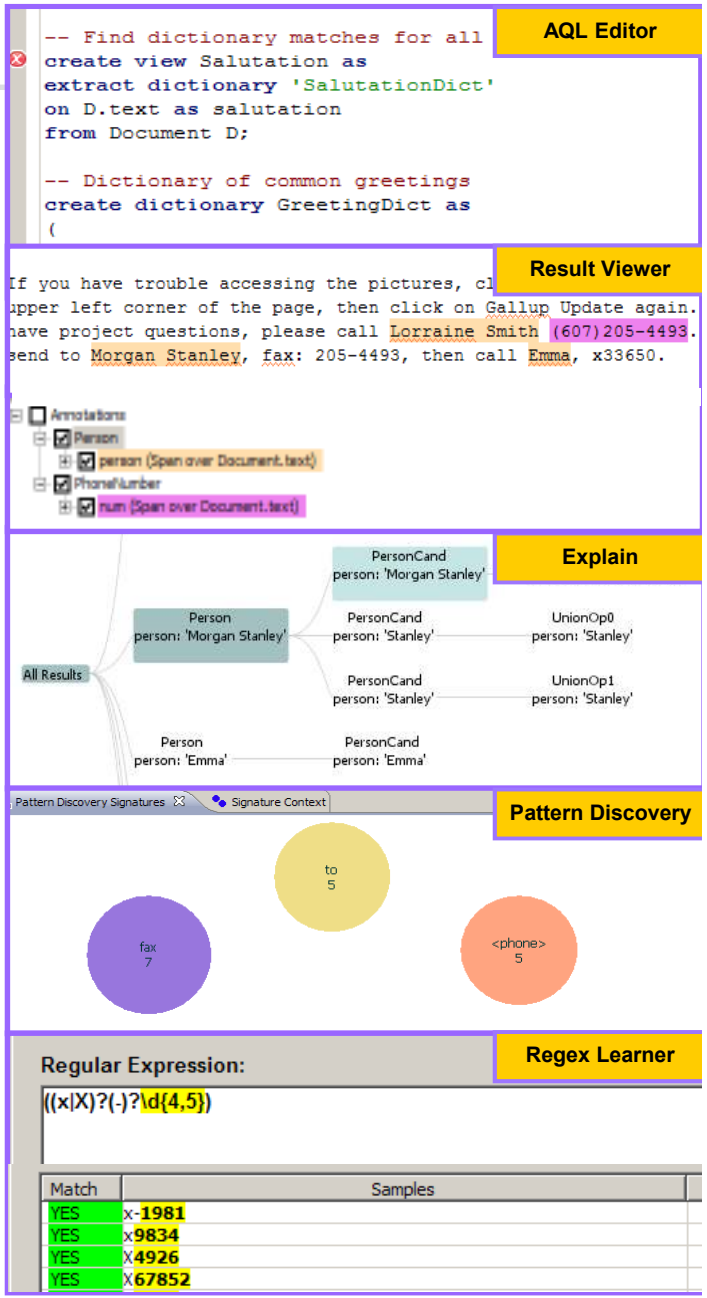
Automatic Discovery

Regex Generator: generate regular expressions from examples

Pattern Discovery: identify patterns in the data

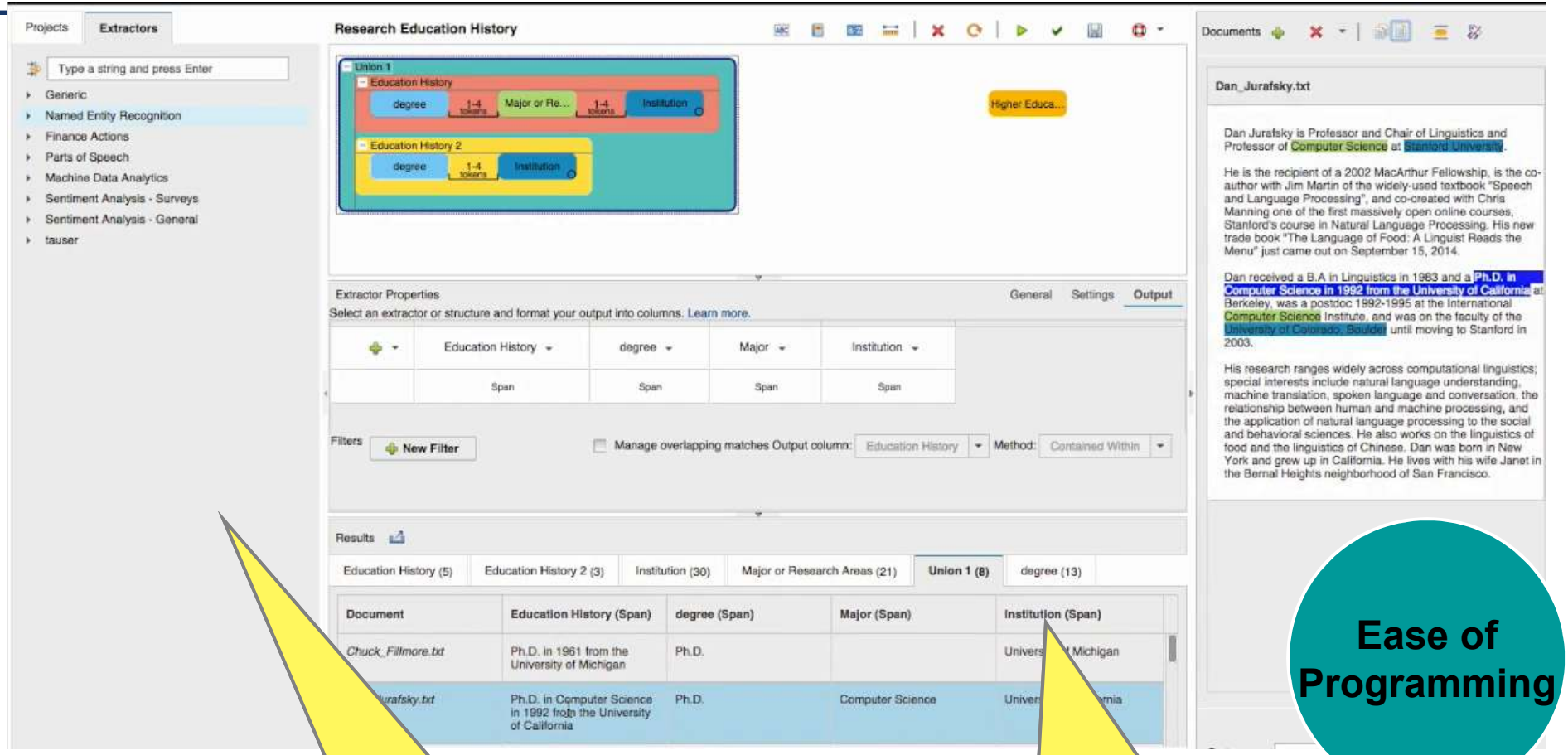
Performance Tuning

Profiler: identify performance bottlenecks to be hand tuned



[Chiticariu et al., SIGMOD '11, Li et al. ACL'12]

Web Tools Overview



Ease of Programming

Ease of Sharing

Concept catalog: share concepts

Project: share extractor development

Canvas: Visual construction of extractors, Customization of existing extractors

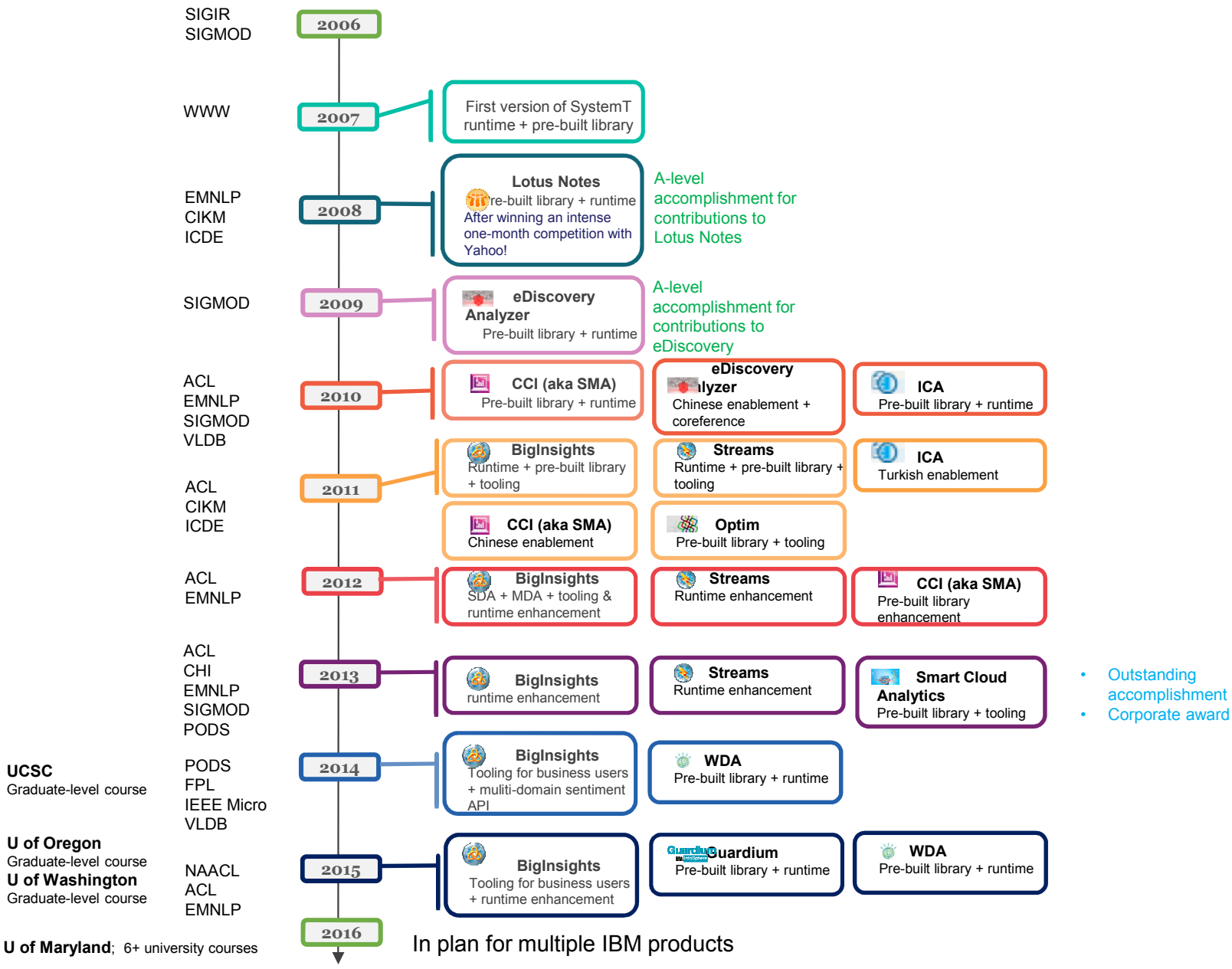
Result Viewer: visualize/compare/evaluate

[Li et al., VLDB'15]

Outline

- Emerging Applications: Case Studies
- Enterprise Requirements
- SystemT : an Algebraic approach
  - Support Enterprise Information
  - Multilingual Support
  - Machine Learning
  - Tooling
- SystemT Courses
- Summary

SystemT Internal & External Impact



Example SystemT Courses

- **University of Washington**
  - LING 575 - Declarative Information Extraction
    - Audience: ~15 graduate students in CLMS program (Professional MS in Computational Linguistics)
    - Prerequisites:
      - LING 570
      - LING 572, Java, and Eclipse are a big plus, but not required
- **University of Oregon**
  - CIS607 - Natural Language Processing and Information Extraction
    - Audience: ~15 graduate students in CIS program (Computer and Information Science)
    - Prerequisites:
      - None. Basic knowledge of AI will be helpful
- **Support from Almaden**:
  - All lecture materials (adapted by the professors for the classes):
  - 1 lecture in person / by Telepresence / Skeype + 1 – 2 office hours
  - Email supports available throughout the course (mainly needed during the first month)
- **Coming soon:**
  - SystemT MOOC (slides/video/story board done)

Outcome

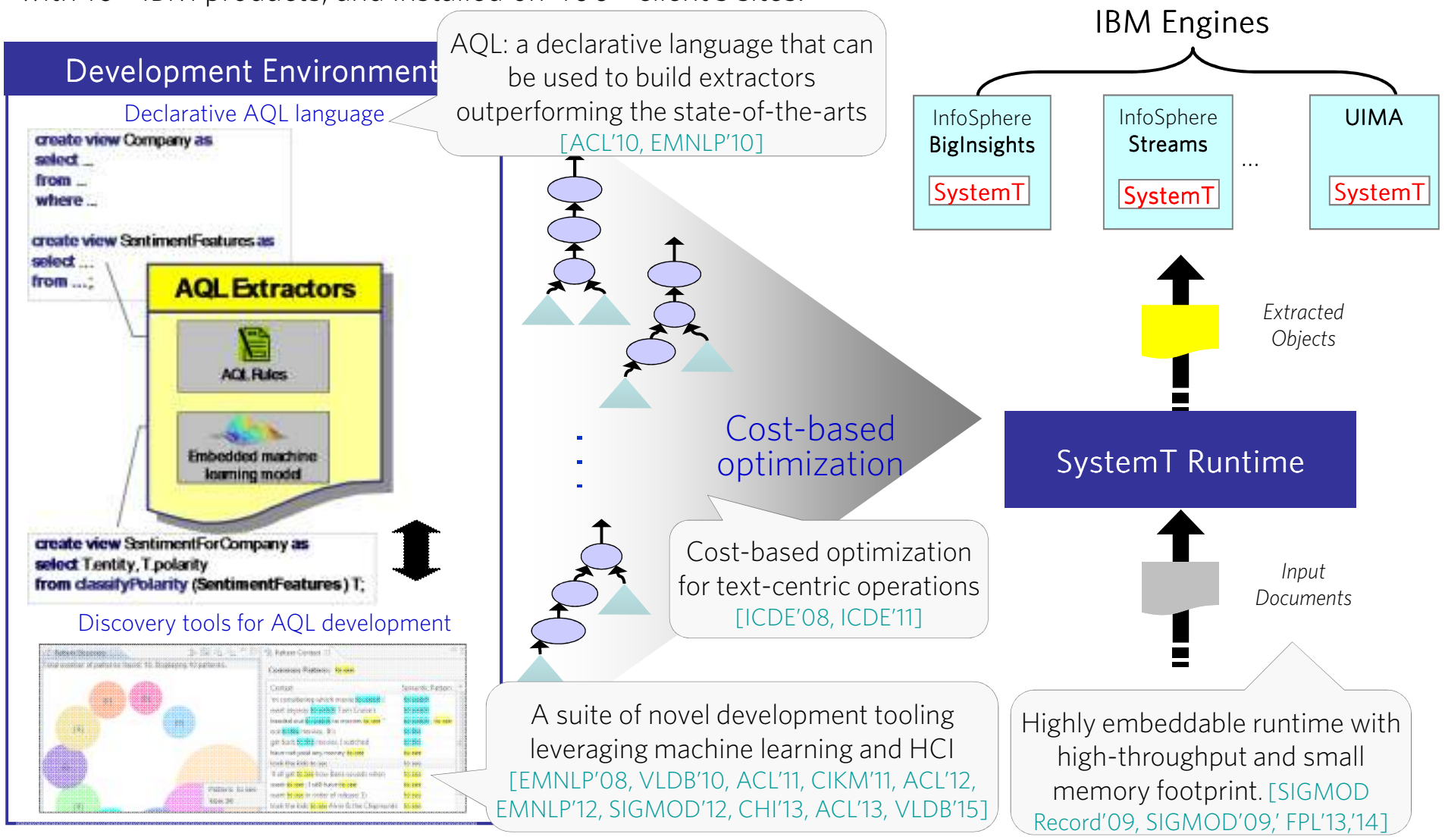
- **University of Washington**
  - 9 group projects: *8 based on / related to SystemT*
    - Extract of Personally Identifiable Information from Email Correspondence
    - Extract sentiment from movie reviews with Extracted Named Entities
    - Apply IE for analyzing learner corpora
    - Extract opinion from customer reviews
    - OrionsBelt: Integrating Sentiment Analysis augmented with Fuzzy Matching into SystemT
    - Extract poetic similes from the body of a poem and then related back to the poem title and author + Extract the poet's lifespan and gender
    - Game of Thrones: Character Relation Extraction
      - Extract family relation + post-processing (de-duplicate + anaphora resolution)
    - DIFFRENT: Differentiating Fiction From REality iN systemT
      - Extract (actor, character) pairs from Wikipedia
      - Extract grammatical constructions that indicate which candidate entities participate in the relations of interest.

Outcome – cont.

- **University of Oregon**
  - 7 student projects + paper presentation by other students
    - Extract bios from Wikipedia
    - Extract information about the native habitat for the IUCN list of endangered mammals
      - The global location of the native habitat, typically in relation to a country or region
      - The terrain that the animal typically lives in, including altitude and the type of vegetation or specific setting the animal is found in or near
  - Extract financial information from financial documents (Company name / performance / profit/ loss / action / expenses / revenue / sales, etc.)
  - Extract financial information from news reports
    - Sentiment / Company actions / performance / profit / loss
  - Extract restaurant reviews that critiqued the service of the restaurant and the décor
  - Extract company name and # of patents granted for the company
  - Extract address and phone # for restaurants at a certain location

Summary

A declarative information extraction system with cost-based optimization, high-performance runtime and novel development tooling based on solid theoretical foundation [PODS'13, 14]. Shipping with 10+ IBM products, and installed on 400+ client's sites.



Find Out More about SystemT!

https://ibm.biz/BdF4GQ

Featured research | Cognitive computing | Client programs | Locations | Our people | Careers

feedback

SystemT

Join/Edit Group

Overview | Publications | Annotated Publications | News | Get SystemT | Educators | Demo

We are hiring! Multiple positions available. Email your resume to Laura Chiticariu (first 5 letters of last name {at} us.ibm.com)

More Information

- Documentation

Research Areas

- Computer Science
- Natural Language Processing
- Data Management

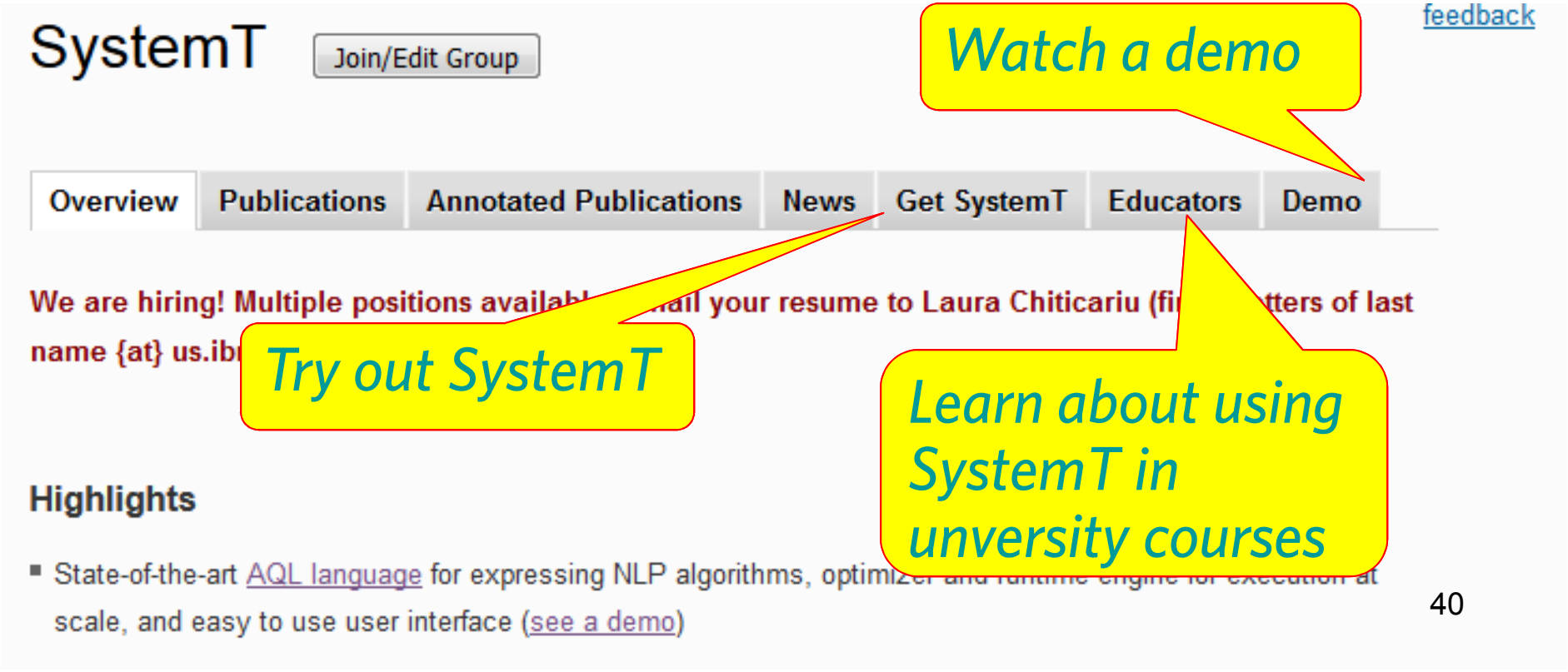
Awards

Highlights

- State-of-the-art AQL language for expressing NLP algorithms, optimizer and runtime engine for execution at scale, and easy to use user interface (see a demo)
- Publications in top NLP, database systems, hardware and HCI conferences
- Currently taught in multiple universities
- Winner of multiple IBM Corporate Awards for its contributions to IBM products and clients

Find Out More about SystemT!

https://ibm.biz/BdF4GQ



Thank you!

- For more information…
  - Visit our website:
    http://ibm.co/1Cdm1Mj
  - Visit BigInsights Knowledge Center:
    http://ibm.co/1DIouEv
  - Learning at your own pace: With the lab materials
  - Contact me
    - yunyaoli@us.ibm.com

SystemT Team:
Alan Akbik, Laura Chiticariu, Marina Danilevsky, Howard Ho, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, Shivakumar Vaithyanathan, Huaiyu Zhu