Meta-Algorithmic Approaches to Semantic Computing

Steven J Simske

Tuesday, 17 June, 2014

OUTLINE

Semantics

Meta-Algorithmics

Examples

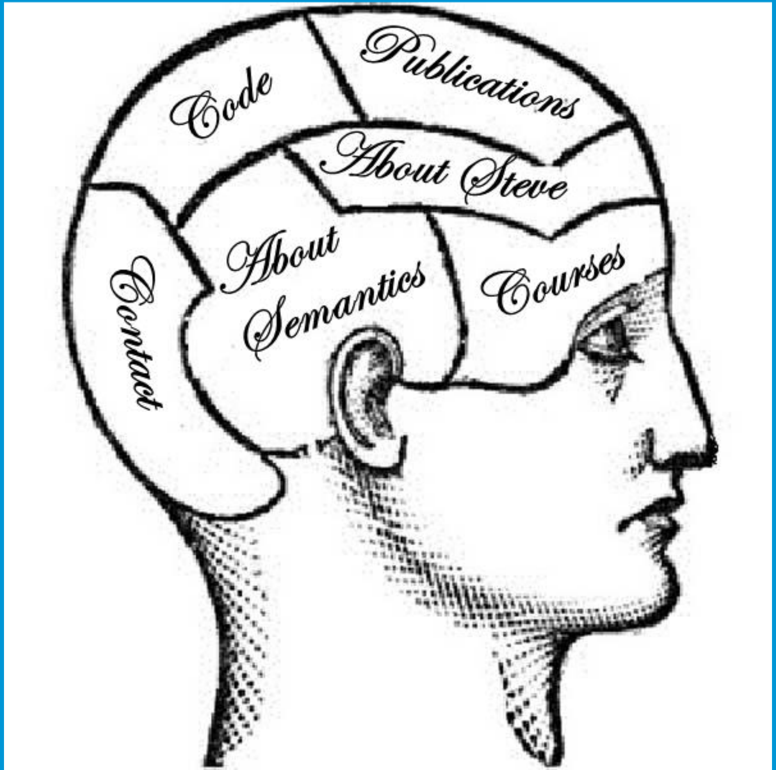
- Synonymic Search
- Document Classification

Functional Means of Optimizing Systems

- Proof by Task Completion
- Sensitivity Analysis
- Translation

The Future

Semantics



http://www.semantics.org/

It’s All Just Semantics

User Intention, Meaning from Media, Mapping Text to Tasks

Meaning

- Beyond analytics
- Beyond correlation

The Value of Figures of Speech

- Hallmark of literary competence: prose, poetry, playwright prowess
- Set yourself a hard challenge and be able to back off to useful, robust solutions

Analysis

- Categorization
- Classification
- Summarization
- Search
- Authentication

http://timespentatsea.blogspot.com/2011/01/mini-cruise-on-sailing-clipper-around.html

Semantic Challenge: Identifying Figures of Speech

Synonymic Figures of Speech

Synecdoche

- Part for the whole, or whole for the part
- Specific for the general, general for the specific
- “Five sail” for “five ships”, an “Einstein” for a brilliant person

Kenning

- Substitute phrase for another phrase
- Figurative expression replacing a noun
- “Book worm” for “Avid reader”, “Wave traveler” for “boat”

Metonymy

- Substitute name, term, expression for closely related name, term, expression
- “The bottle” for “strong drink”, “Wall Street” for “financial industry”

http://timespentatsea.blogspot.com/2011/01/mini-cruise-on-sailing-clipper-around.html

Boolean Figures of Speech

Litotes

- Affirmation through Negating its Opposite
- A = Not(Not(A))
- “He is no bad programmer”, “That is not bad!”

Chiasmus

- Reversing order of paired expressions
- A B B A
- “Evil never rests, and resting is never evil”

Antithesis

- Juxtaposition of contrasting ideas
- A = Not(B)
- "Love is an ideal thing, marriage a real thing." *Goethe*

A^1 Whoever *sheds*
B^1 the *blood*
C^1 of *man*
C^2 by *man*
B^2 will his *blood*
A^2 be *shed.*

http://azure.org.il/include/print.php?id=579

Figures of Speech with Special Word Usage

Symbolic Substitution, Equivalent Meaning

Irony, Sarcasm, Satire

- The US Secret Service wants someone to build a sarcasm detector for Twitter:
- http://www.independent.co.uk/life-style/gadgets-and-tech/the-us-secret-service-wants-someone-to-build-a-sarcasm-detector-for-twitter-9493049.html
- "Despite the obvious benefits of government agents being able to sort credible threats from the merely facetious, recognizing sarcasm – or most nuanced moods – has so far proved difficult for computers"

Metaphor

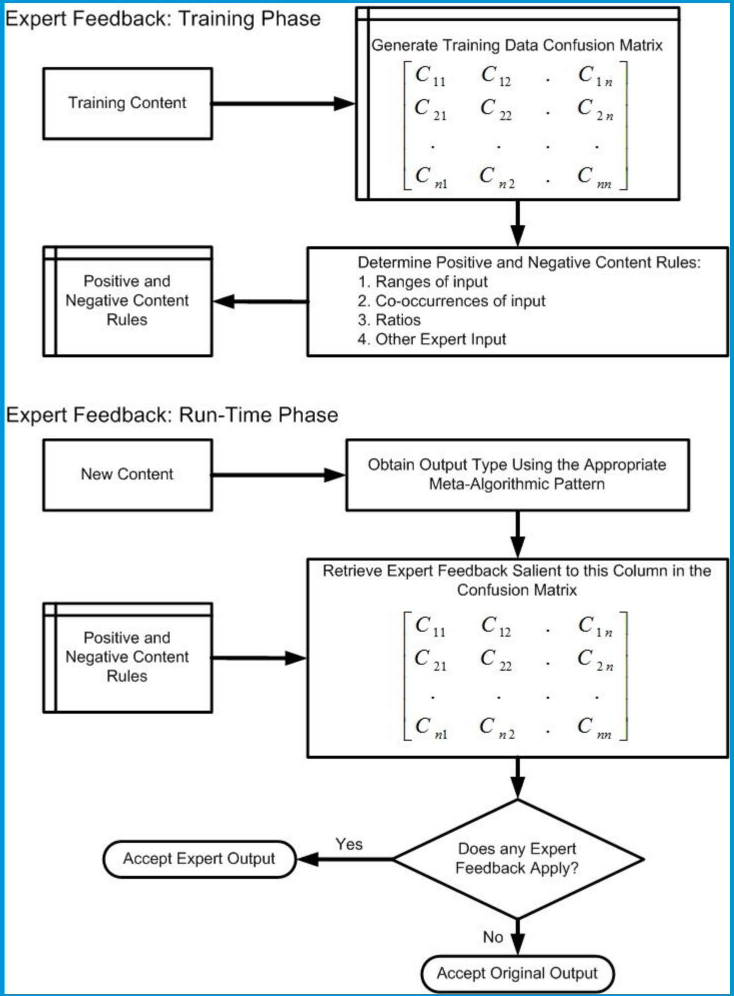
- A phrase or term is applied to something figuratively
- "The guy giving this keynote is a dinosaur"

Simile

- One of the easier-to-find figures of speech: the cues are "like" and "as"
- Can be used to generate associates for other parts of speech: metaphor, synecdoche, etc.

http://weblogs.baltimoresun.com/news/technology/2010/05/an_online_sarcasm_detector_yea.html

Meta-Algorithmics



When the Apple Falls out of the Sky

Part One

Want to Invent Mechanics, the Laws of Motion, the Universal Law of Gravitation, the Calculus, the Reflecting Microscope and Optics?

Well you can't!

They've already been discovered

But Newton did show that for every action, there is an equal but opposite reaction

- You will not discover the laws of the universe
- You will have to use two or more of them to build a larger, more integrated system

Image from http://thefishbowlnetwork.com/blog/the-fruit-that-makes-history/

When the Apple Falls out of the Sky

Part Two

Did Apple Invent the PC? The Phone? The Stereo? The Internet? The Browser?

No, they didn't!

They'd already been discovered

But Apple did show that for every approach, there is a way to integrate it into an overall system – there are at least four lessons:

- Nothing new invented
- Everything already invented readily integrated
- Hybridizing multiple functions is not additive, it is exponential
- With the right platform, you can get others to do your work

What are the major trends?

The cloud, mobile, big data, near-infinite storage, massive computational power

Unprecedented Data Growth

Focus on Immediate Data

- Social Internet
- Sensors
- Sentiment Analysis

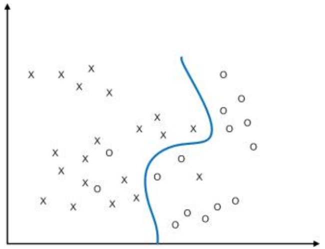
Machine Intelligence

- Predictive Analysis
- Clustering
- Categorization
- Classification
- Data Mining & Data Insight

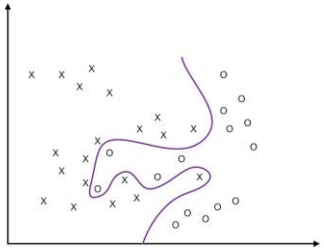
Inching Closer to Exhaustive Search

One Size Does Not Fit All

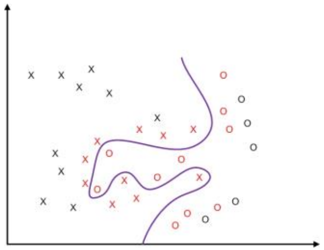
There are many different ways to achieve machine intelligence



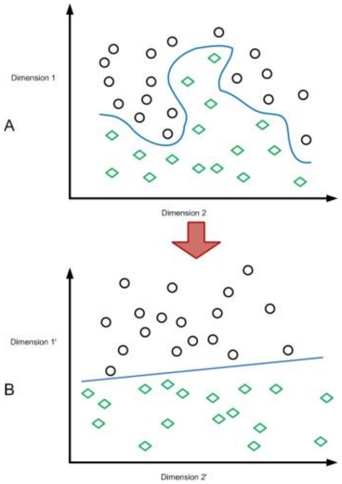
Simple Boundary



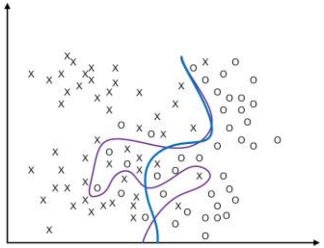
Complex Boundary



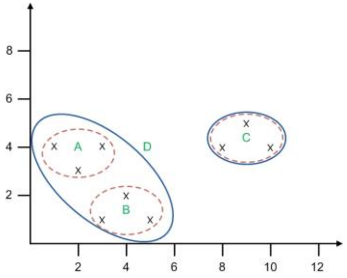
Support Vector Defining Samples in Red



Original and Transformed Space



Often the Simpler is More Robust



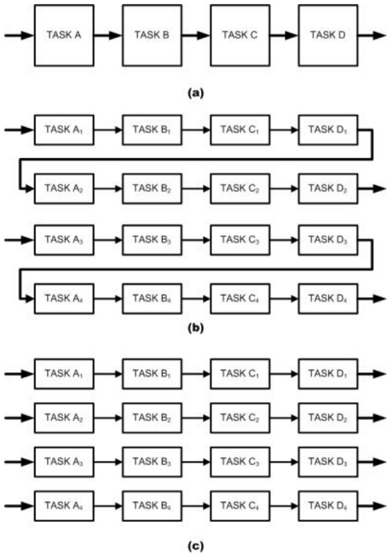
Automatic Cluster Determination (F-Score Guided K-Means)

Parallelism by Task

One or more sequential steps is restructured to be parallel

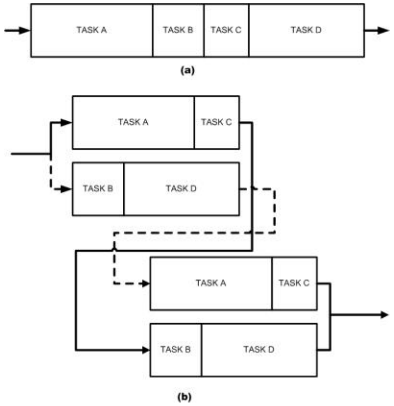
Queues

Subtasks are still performed sequentially



Variable-Sequence

Re-crafts sub-pipelines based on dependency mapping – here D dependent on B but not A and C dependent on A but not B



May be more robust to "hanging" on a long sub-process but requires more domain knowledge and may prevent new, more efficient, dependencies

Parallelism by Component

Reconstruction of a data set so that two or more sub-data sets can be processed in parallel

Modified Amdahl's Law

$$t_{overall} = k_{BO} k_{SchO} k_{SRfO} (1 - P + \frac{P}{N}) t_{serial}$$

Where

k_{BO} is the branching overhead factor

k_{SchO} is the scheduling overhead factor

k_{SRfO} is the structural reframing overhead factor

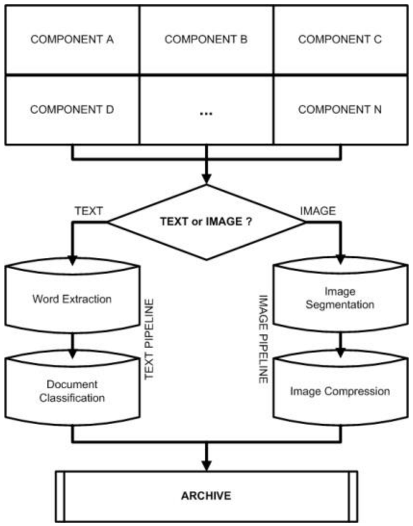
P=percent of the image processing that can be made parallel

N=number of parallel processors available

$$t_{overall} = \frac{k_{BO} k_{SchO} k_{SRfO} (1 - P + \frac{P}{N})}{N^{2T-2}} t_{serial}$$

Loops are unwrapped into branches; examples:

- Split-and-merge, such as mixed media processing where different media are assigned to different (optimized) branches
- Componentization through decomposition (in imaging by processing the color planes separately)
- Model down-sampling (processing of smaller data sets until the salient information is extracted)



Parallelism or Preparing for Parallelism?

Which is more important depends on the system resources and the type of task

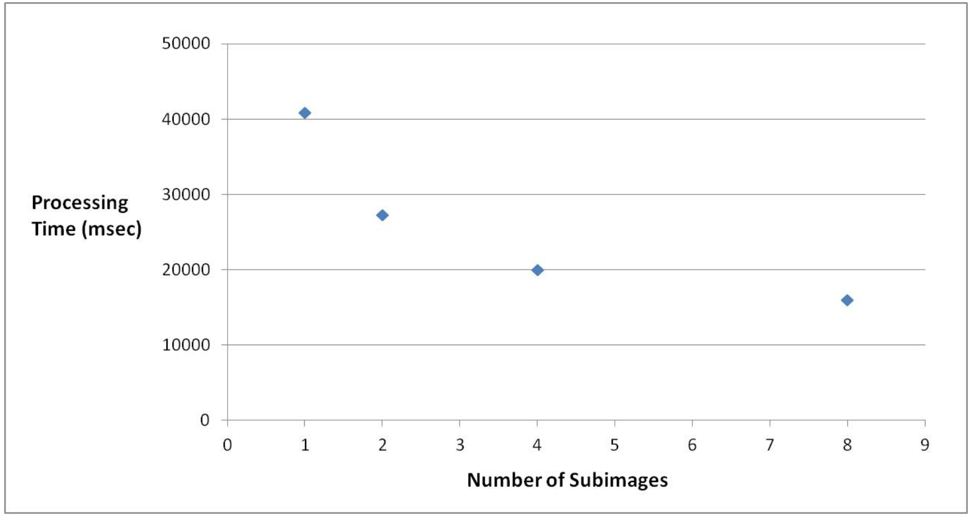
The simple preparation of data for parallel processing (*structural re-framing*) is comparable to the effective parallelism of simple parallel processing

$$EP = n_{subtasks}\left(\frac{t_{proc}(mean)}{t_{proc}(\max)}\right)$$

EP=Effective Parallelism and is limited by the longest subtask processing time

$$\%TI = \left(\frac{EP-1}{EP}\right)\cdot 100\%$$

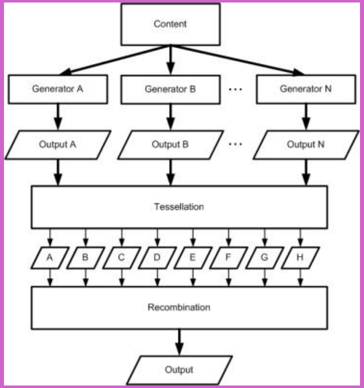
%TI= Percent Throughput Improvement

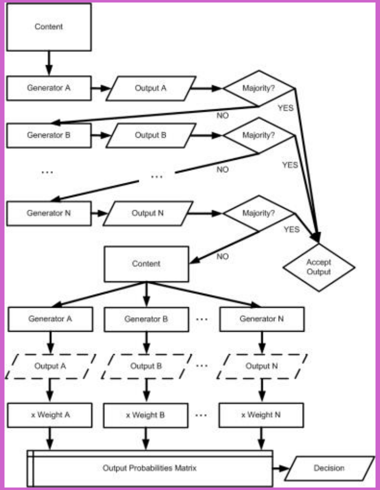


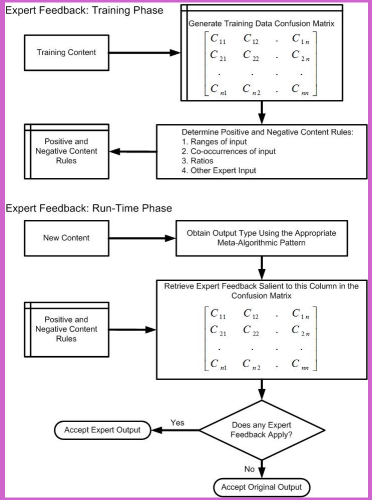
Irrespective of EP and %TI, just making the subtasks smaller in memory and complexity (structural re-framing) can lead to substantial improvement in throughput, *even when no parallel processing is used.*

Meta-Algorithmics

- First Order
- Second Order
- Third Order

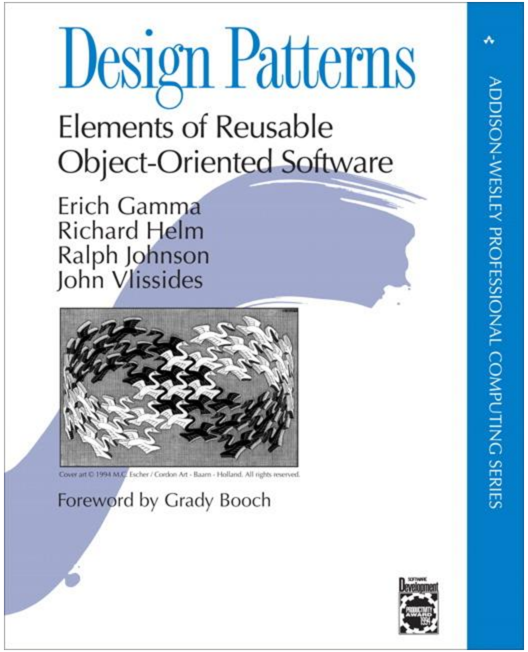


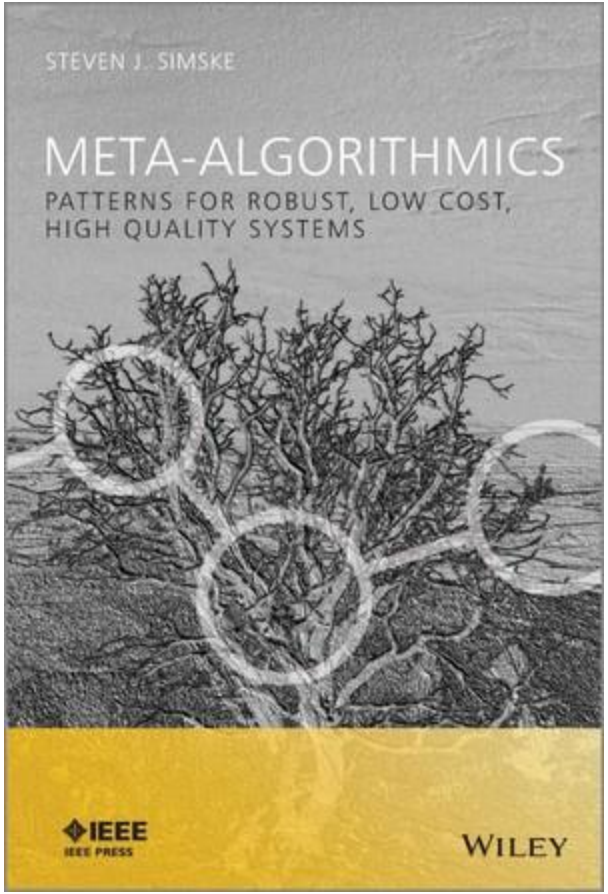




What is needed?

A set of design patterns to help the designers of intelligent systems, akin to OOS designs

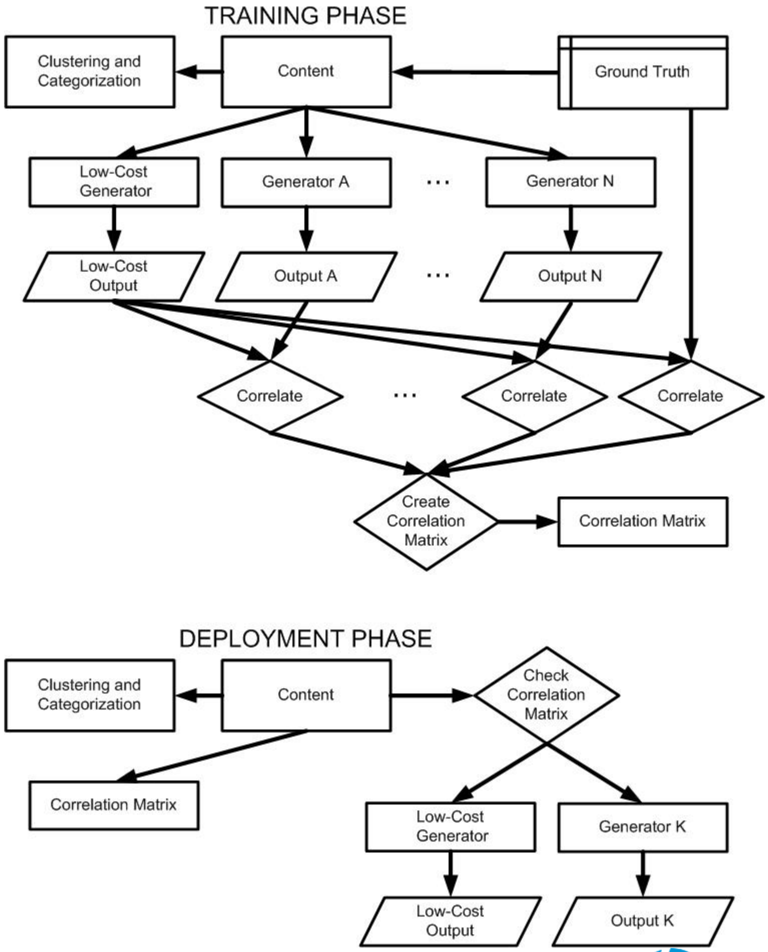




First Order Patterns

These patterns include the simplest possible means of combining two or more generators

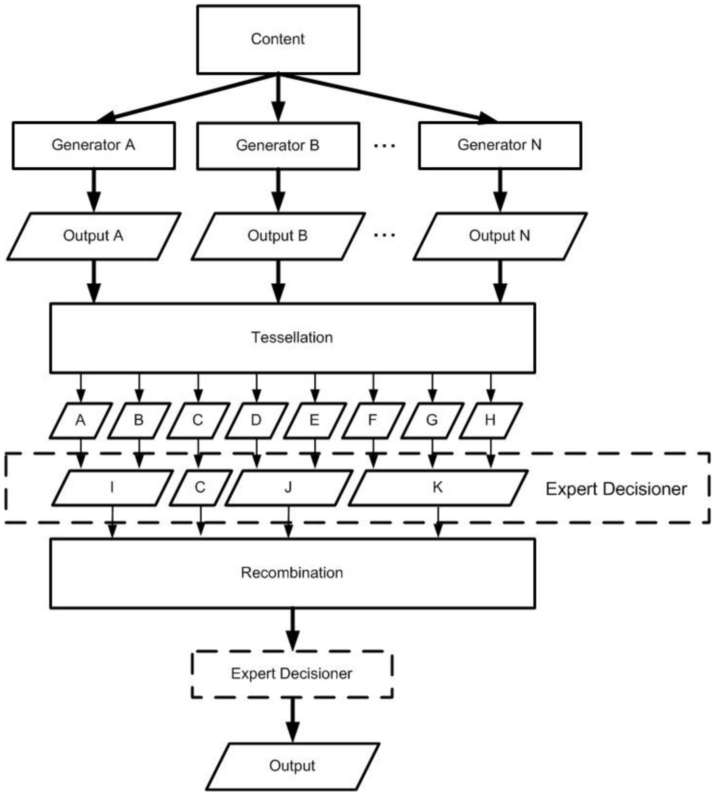
1. **Sequential Try and Try pattern, using serialized attempts to achieve a given result**
2. **<u>Constrained Substitute</u>, using simple substitution**
3. **Voting and Weighted Voting pattern, the simplest combination**
4. **Predictive Select and Tessellation and Recombination, requires some domain expertise since it requires de-aggregating the input space into a useful set of classes**
5. **Tessellation and Recombination requires de-aggregating the data itself during the tessellation operation, and also usually requires domain expertise**



Second Order Patterns

These patterns include the simplest possible means of combining two or more generators

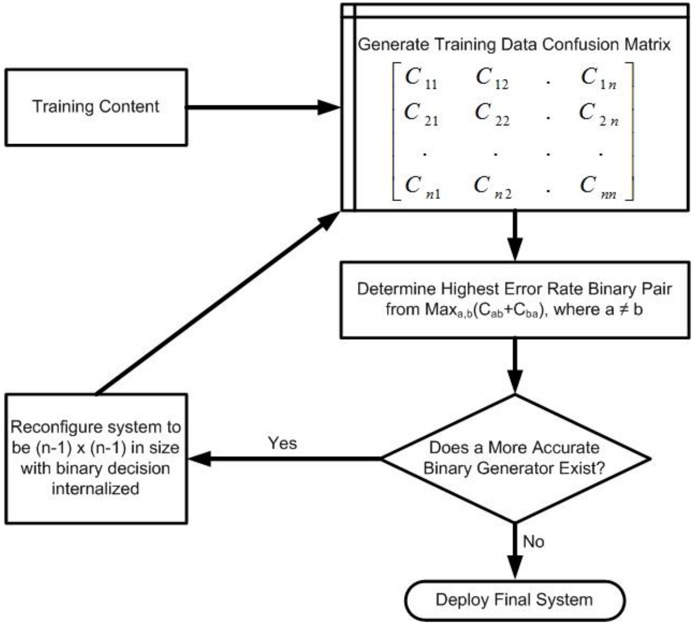
1. **Confusion Matrix and Weighted Confusion Matrix**
2. **Confusion Matrix with Output/Probability Space Transformation**
3. **<u>Tessellation and Recombination with Expert Decisioner</u>**
4. **Predictive Selection with Secondary Engines**
5. **Single Engine with Required Precision**
6. **Majority Voting or Weighted Confusion Matrix**
7. **Majority Voting or Best Engine**
8. **Best Engine with Differential Confidence or Second Best Engine**
9. **Best Engine with Absolute Confidence or Weighted Confusion Matrix**



Third Order Patterns

These patterns use output characteristics to improve the system behavior. Feedback involves corrective action on the system settings and sensitivity analysis looks for the best places to decimate the input further

1. Feedback
2. Proof by Task Completion
3. Confusion Matrix for Feedback
4. Expert Feedback
5. Sensitivity Analysis
6. Regional Optimization or Extended Predictive Selection
7. Generalized Hybridization



Meta-Algorithmics, the 3rd form of parallel processing

Parallelism by Architecture, Approach, Analysis and Algorithm

Extended: Meta-Algorithmics

Architectural Variations

- Series
- Parallel
- Generalized series-parallel

Approach

- Using multiple algorithms to fully explore the input space

Analysis

- Confusion matrix
- Feedback

Algorithms

- Predictive algorithms
- Sequential algorithms
- Validated /feedback approaches

The Opportunity is Huge

Is the guy/gal next to you smarter than you?

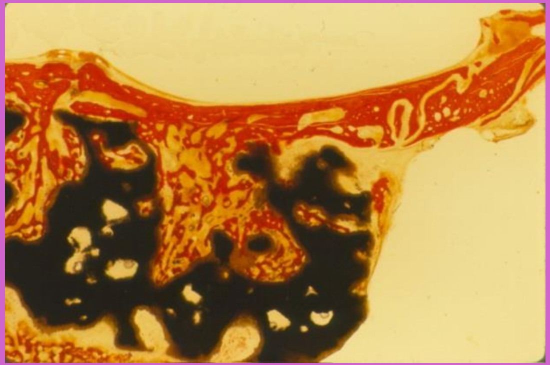
- He/she is 50% of the time
- 100% of the time, he/she is smarter than you at something

Wouldn't you like to have your worst enemy's latest accomplishment make you even better?

- Every improvement can build up your overall system's performance, accuracy, cost and/or robustness

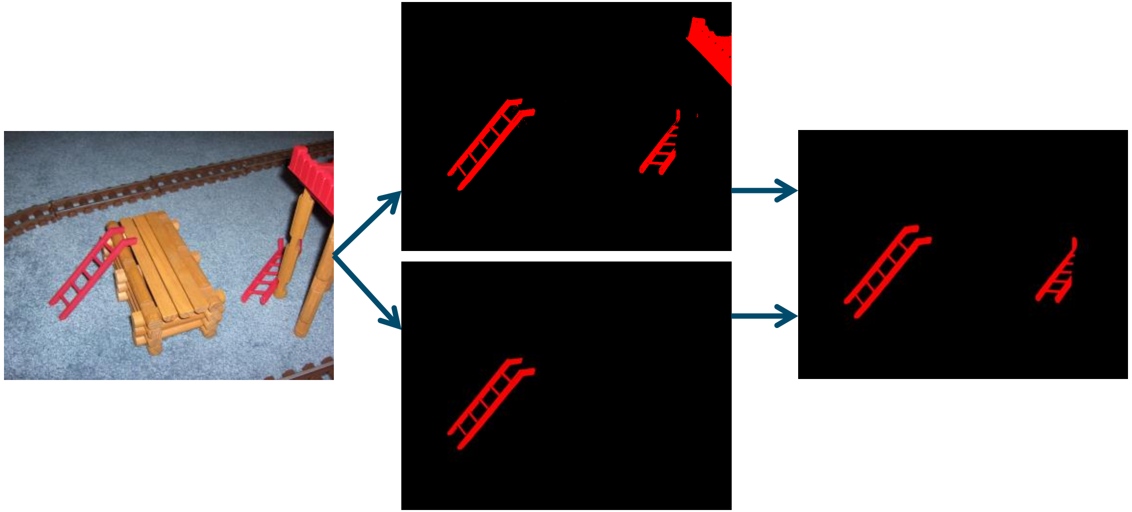
http://matthewblakewilliams.com/on-making-enemies/ http://neverbetter.com/Love-Your-Enemies-And-You-Won/t-Have-Any-Bumper-Stickers?cPath=21_26_44&

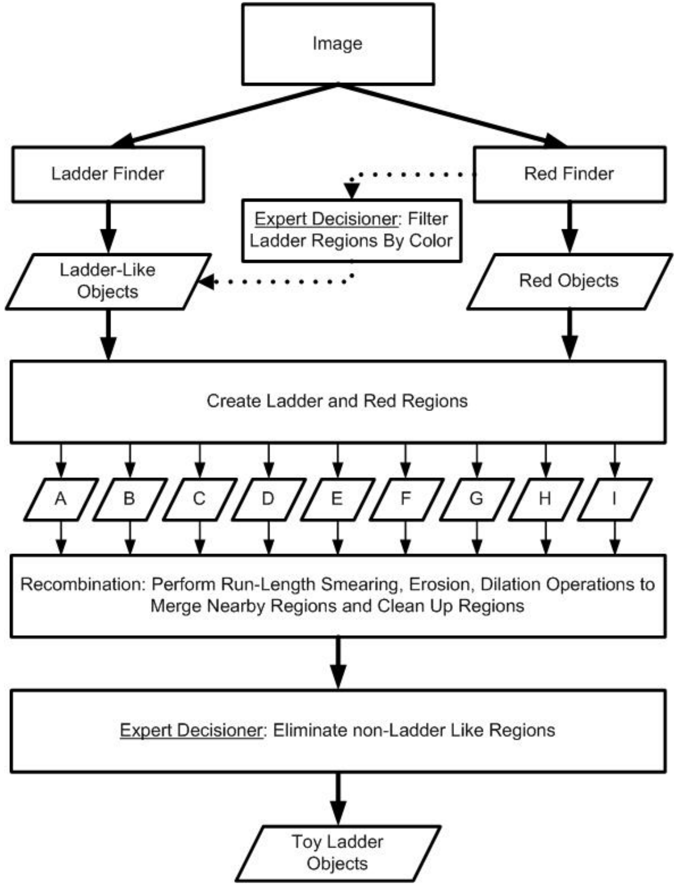
Applications



Sample Robust-Optimization

Tessellation and Recombination with Expert Decisioner





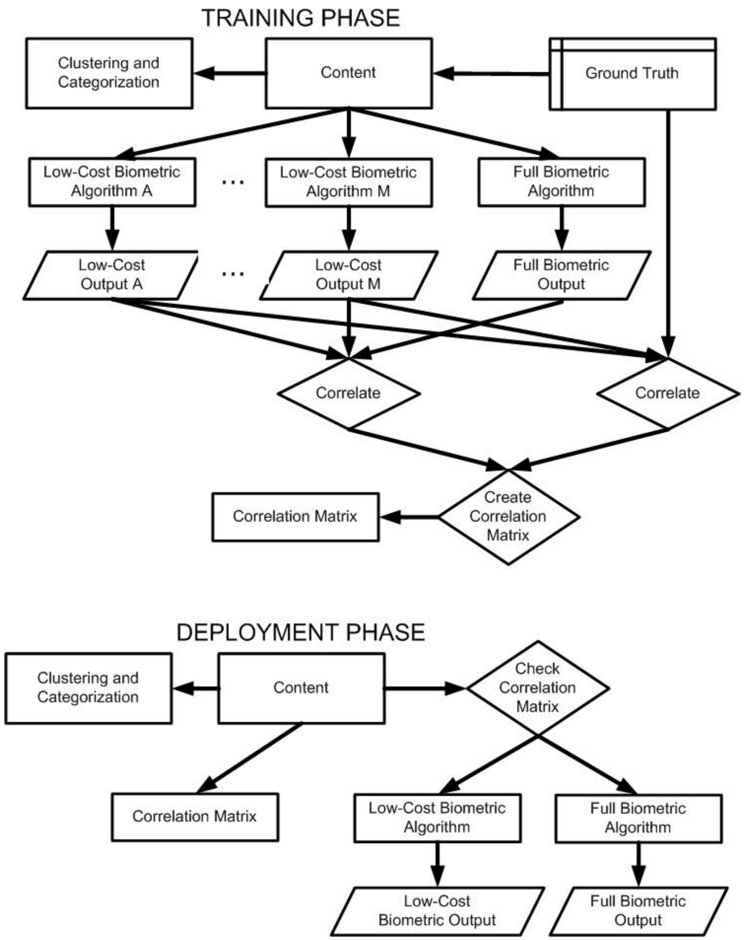
Expert Decisioner:

$200 \leq r \leq 255$ $0 \leq g \leq 120$ $0 \leq b \leq 120$

Sample Cost-Optimization

Constrained Substitute

	Biometric Algorithm			
Datum	Low-Cost A	Low-Cost B	Low-Cost C	Full Algorithm
Processing Time, t_p (msec)	110	170	140	250
#Times to Process 2.0 sec of Voice Data to Obtain $p<10^{-9}$	8	5	8	4
#Times to Process 2.0 sec of Voice Data to Obtain $p<10^{-6}$	5	4	5	2
#Times to Process 2.0 sec of Voice Data to Obtain $p<10^{-3}$	2	3	3	1
Processing time for $p<10^{-9}$ (msec)	880	850	1120	1000
Processing time for $p<10^{-6}$ (msec)	550	680	700	500
Processing time for $p<10^{-3}$ (msec)	220	510	420	250

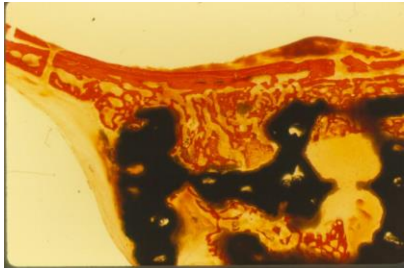


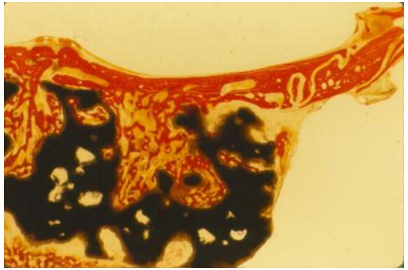
Sample Quality-Optimization

Predictive Selection

Three separate image analyzers were used to classify each of the biomedical images:

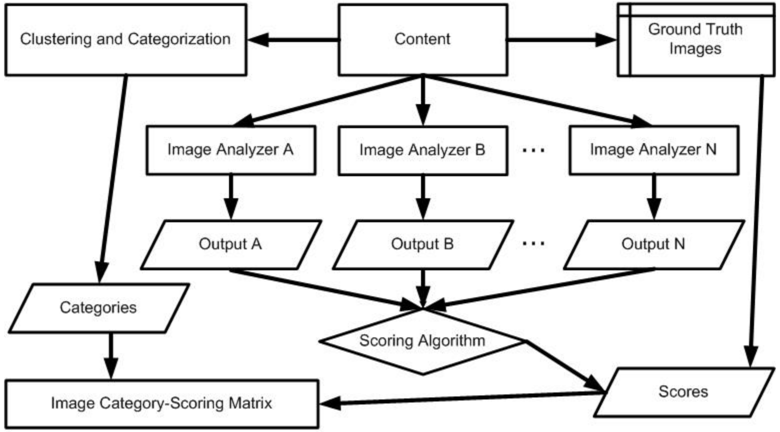
(1) excellent ingrowth and excellent apposition
(2) poor ingrowth and excellent apposition
(3) excellent ingrowth and poor apposition
(4) poor ingrowth and poor apposition



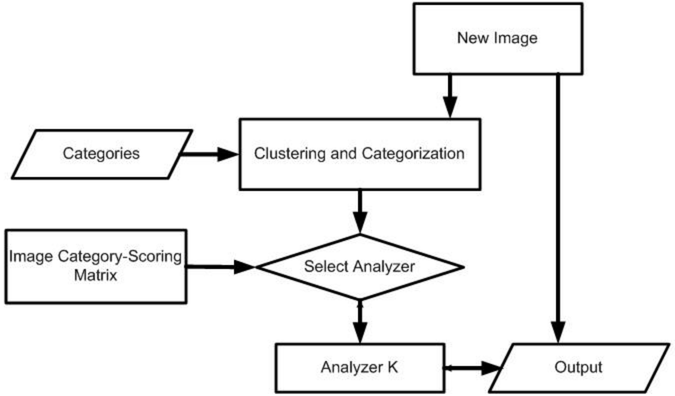


The Predictive Select pattern (choosing the output from the Analyzer with the highest precision for each subrange) improved the accuracy on test images to 0.76 from 0.66 without meta-algorithmics. This is a reduction in error from 34% (the error rate for the best of the three individual Analyzers) to 24%, or a relative reduction in error rate of 29%

Implant Biocompatibility: Statistical Learning Phase



Implant Biocompatibility: Run-Time Phase



Sample Cross-Discipline Results

Domain to which meta-algorithmics were applied	Percent improvement (type of system behavior improvement) over best individual system
Image segmentation (extracting specific objects)	**8.3%** (reduction in error rate)
Biometrics (system to ID an individual with given confidence)	**7.2-10.7%** (reduction in overall system cost to achieve a required level of biometric accuracy)
Medical data storage space (directly proportional to cost)	**94.7%** (reduced storage space with full clinical value of the data retained)
Medical image analysis (quantitative histomorphometry)	**29%** (reduction in error rate)
Optical character recognition	**5.0-27.6%** (reduction in error rate, depending on preferred system costs)
Document processing (document workflow completion)	**>28%** (throughput performance improvement)
Security printing data extraction	**13.0%** (reduction in bit level reading error rate)
Image processing (pipeline)	**54.8%** (reduction in system processing time, or improvement in throughput performance)
Speech recognition (converting audio to text)	**13.3%** (reduction in ASR, automatic speech recognition, error rate)
Document classification	**16.2%** (reduction in system error rate)
Image surveillance (tracking from video)	**13-29%** (reduction in processing time, or improvement in throughput performance)
Text summarization (selecting optimal sets of sentences)	**45%** (reduced error rate)

In each case, multiple meta-algorithmic approaches provided improvement over the best individual algorithm

Meta-Algorithmics, A Recap

They Follow the Four Lessons Learned

1. Nothing New Invented

- Continuum with ensemble and voting methods
- Systematizes the architectural needs for intelligent systems

2. Everything Already Invented Readily Integrated

- Existing algorithms, intelligence engines and systems readily incorporated into engines
- Domain knowledge valuable, but deep domain knowledge not, usually, necessary

3. Hybridizing Multiple Functions is Not Additive, It Is Exponential

- Each pattern allows a unique exploration of different combinations, and different types of combinations, of the individual intelligence engines
- There are far more meta-algorithmic systems than the individual systems

4. With the Right Platform, You Can Get Others to Do Your Work

- With a meta-algorithmic, every new, better single algorithm can add to your system
- The better your competition, the better your system

The Future of Meta-Algorithmics

Next steps and scaled-up systems…

Other Fields

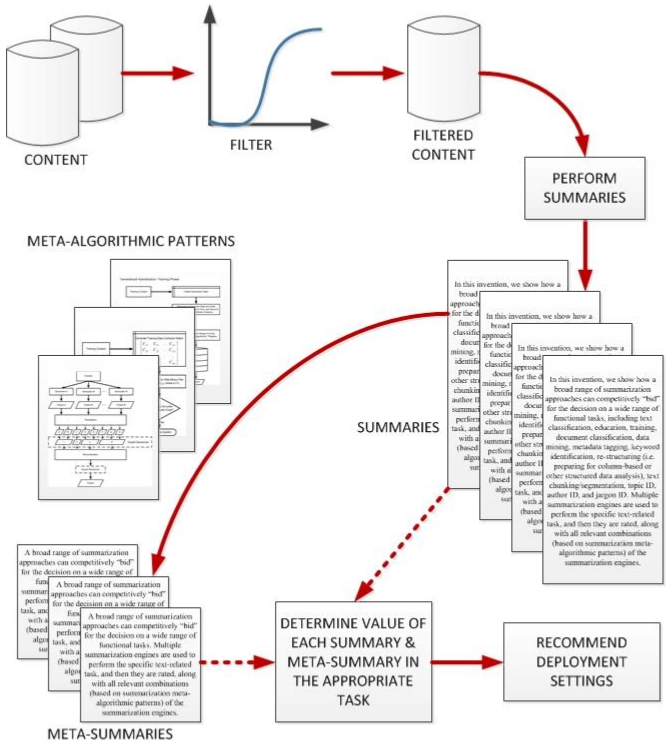
- Analytics
- Data mining
- Expertise-Driven Actionable Analytics

Additional automation

- Many of the experiments performed to date compare the meta-algorithmic approaches after the fact: automating the selection of the best meta-algorithmic pattern is a next step

Larger system design

- Using meta-algorithmics as a means of generating candidate systems automatically
- Example (see right): functional summarization uses meta-algorithmics to select filtered system input



Semantics

Synonymic Search
Document Classification

http://www.onelink.net/Solutions/DocumentManagement/tabid/65/Default.aspx

User interaction with a database

Search is used to access desired content

Search is used to access a database through queries

Automated search of importance to:

- Data mining
- Knowledge generation
- Machine intelligence

Searches comprise:

- Individual text terms
- Boolean expressions
- Multiple search engines

Search can be improved by:

- Extending the search engine capabilities by increasing the likelihood that related documents are found for a particular search query
- Increasing the search efficacy when the user has only a vague idea about what she is trying to find
- Providing a means to optimize, for several factors, how to select documents associated with a query within any corpus

Query Expansion

Expanding a single search query into a series of related searches

Expansion by synonyms

$$N_Q = \prod_{i=1}^{N_S} (S_i + 1)$$

Where

N_Q is the total number of queries

S_i is the number of synonyms for term I

N_S is the number of terms in the query having one or more synonyms (if no synonyms, then this value is simply 1)

The synonymic search set is typically limited to proximate (and not associated) synonyms in order to keep the number of searches manageable

Many commercially-available, freely-available and proprietary synonym lists exist:

WordNet

http://www.synonym-finder.com/

http://www.synonym.com/

To prevent an open-ended number of queries, the total number of queries may be limited to an absolute maximum of, for example, 25 queries

Selecting synonymic search queries

Longer query expansions may need to be pruned to a reasonable set

- Excess searches are pruned based on the relative synonymic relationship between each of the terms.
- Suppose the user types in the query “class list for Stanford”. For the term “class”, the user will get the following synonyms: set, group, division, grade, rank, category, order (etc.).
- For the term “list”, the user will get the following synonyms: catalog, inventory, register, record, roll, directory (etc.).
- Already, the number of possible synonymic queries is 56 (that is, 8 x 7), but no more than 25 are allowed
- The preferred implementation is to have the synonym database structured such that the synonyms are rated for their “closeness” or “proximity” to the original word

```
<OriginalWord proximity=”1.0”>
  <Spelling>class</Spelling>
  <Synonym proximity=”0.9”>set</Synonym>
  <Synonym proximity=”0.85”>group</Synonym>
  <Synonym proximity=”0.72”>division</Synonym>
  <Synonym proximity=”0.65”>grade</Synonym>
  <Synonym proximity=”0.51”>rank</Synonym>
  <Synonym proximity=”0.42”>category</Synonym>
  <Synonym proximity=”0.23”>order</Synonym>
</OriginalWord>
```

Assigning weights based on proximity

Using, for example, equal spacing across the range

- Note that the “weights” or “proximities” defined above can be further weighted/treated by the “semantics” of the query—i.e. if a query asks for a “ball sport” then any synonyms of “ball” denoting “dancing” rather than “sports equipment” should be discarded.
- Such semantic weighting is, in general, quite difficult, and so weighted synonyms such as those demonstrated here help work around this problem.
- Weights can be defined

  (a) manually

  (b) automatically based on the co-occurrence of such terms in web sites, documents, corpuses, etc.

  (c) automatically based on the order the synonyms occur in a linguistic engine such as WordNet.

```
<OriginalWord proximity=”1.0”>
  <Spelling>class</Spelling>
  <Synonym proximity=”0.875”>set</Synonym>
  <Synonym proximity=”0.750”>group</Synonym>
  <Synonym proximity=”0.625”>division</Synonym>
  <Synonym proximity=”0.500”>grade</Synonym>
  <Synonym proximity=”0.375”>rank</Synonym>
  <Synonym proximity=”0.250”>category</Synonym>
  <Synonym proximity=”0.125”>order</Synonym>
<//OriginalWord>
```

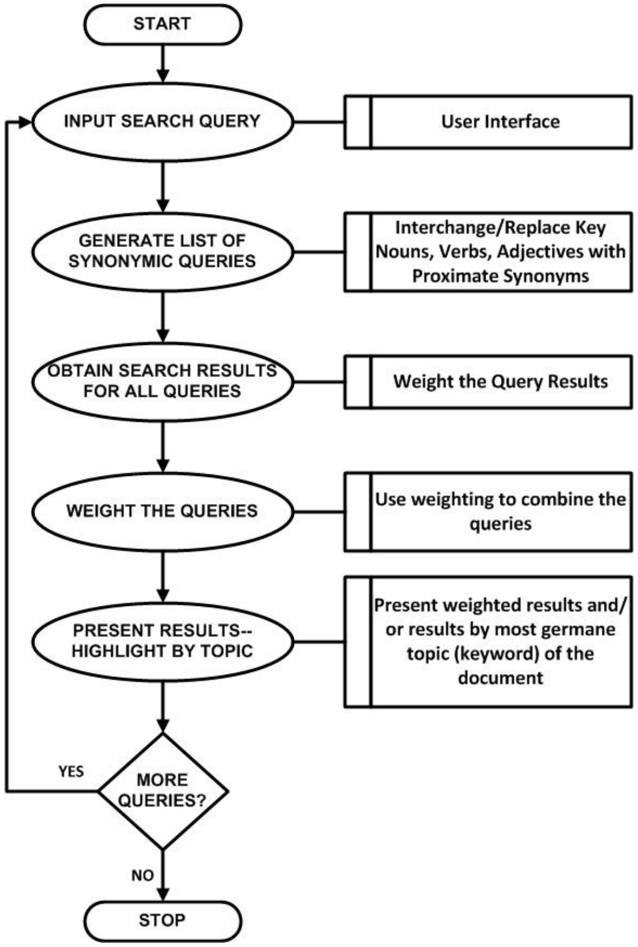
Synonymic Queries

Synonyms: focus on nouns, verbs and adjectives

Weight the query results:

(1) Based on proximity weights

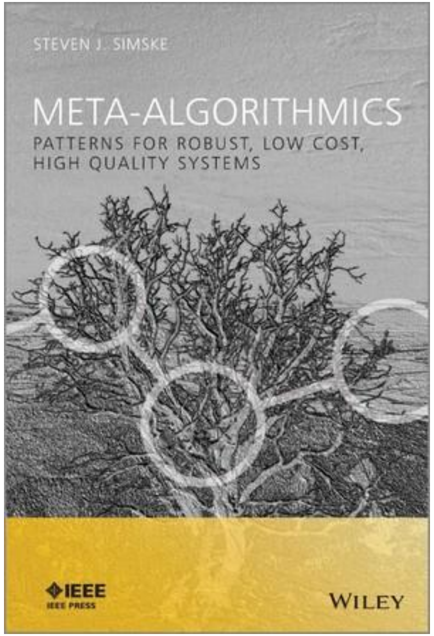
(2) Based on search engine weights (see below)



Using Combinatory (Meta-Algorithmic) Search

Combine 2+ algorithms or systems into a single system for machine intelligence

The meta-algorithmic approach considered is termed "synonymic search", which allows a single search query to be expanded into a set of queries representing synonymic expressions for the original query. The approach also allows tuning of the amount of synonymic broadening to be applied to the received query for constructing the set of synonymic search queries. Identification of resulting documents responsive to each of the plurality of queries is received, and such received documents are ranked based at least in part on a weighting assigned to each of the plurality of queries.



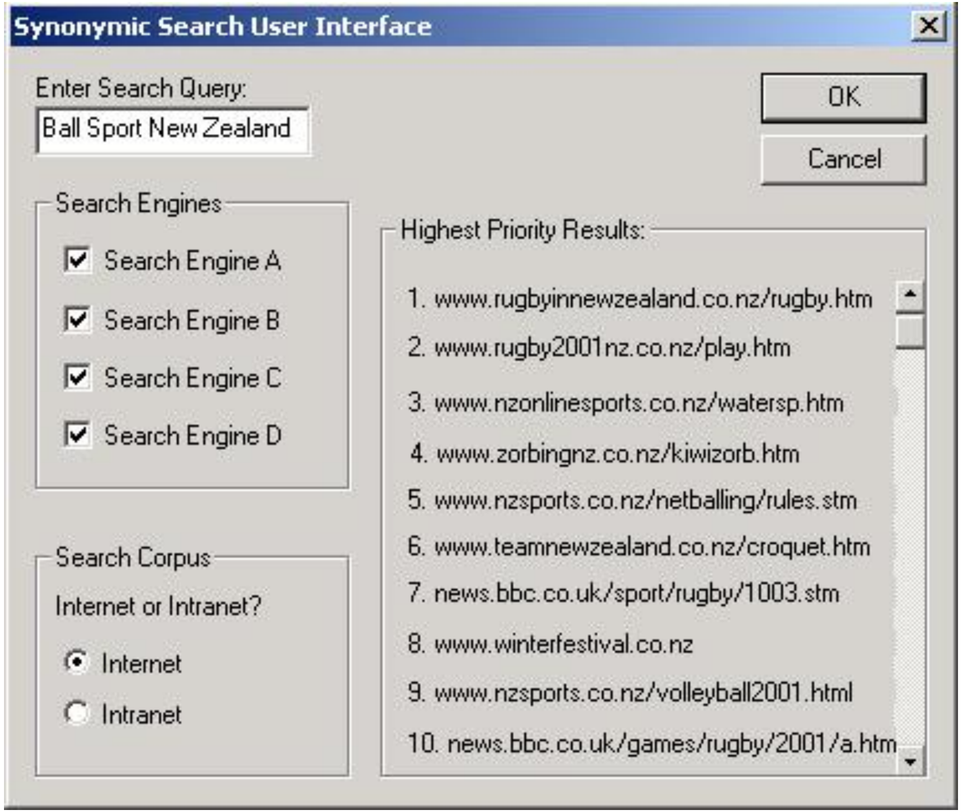
Running Multiple Engines

After all the search queries have been defined, they are performed on one or more search engines (see figure).

All of the queries are provided as input for the search engine and the search engine returns the web sites, documents, etc. that it determines to be best matches. These matches are typically presented in order of relevance, utility, hit frequency, or other reasonable metric, and are presented to the user ranked from 1 to M, where M is the number of “hits” or “matching pages” found.

(1) the engines themselves may be weighted by the confidence in the engines

(2) the order of the results may be weighted, according to their rank in the output set provided by the search engine.



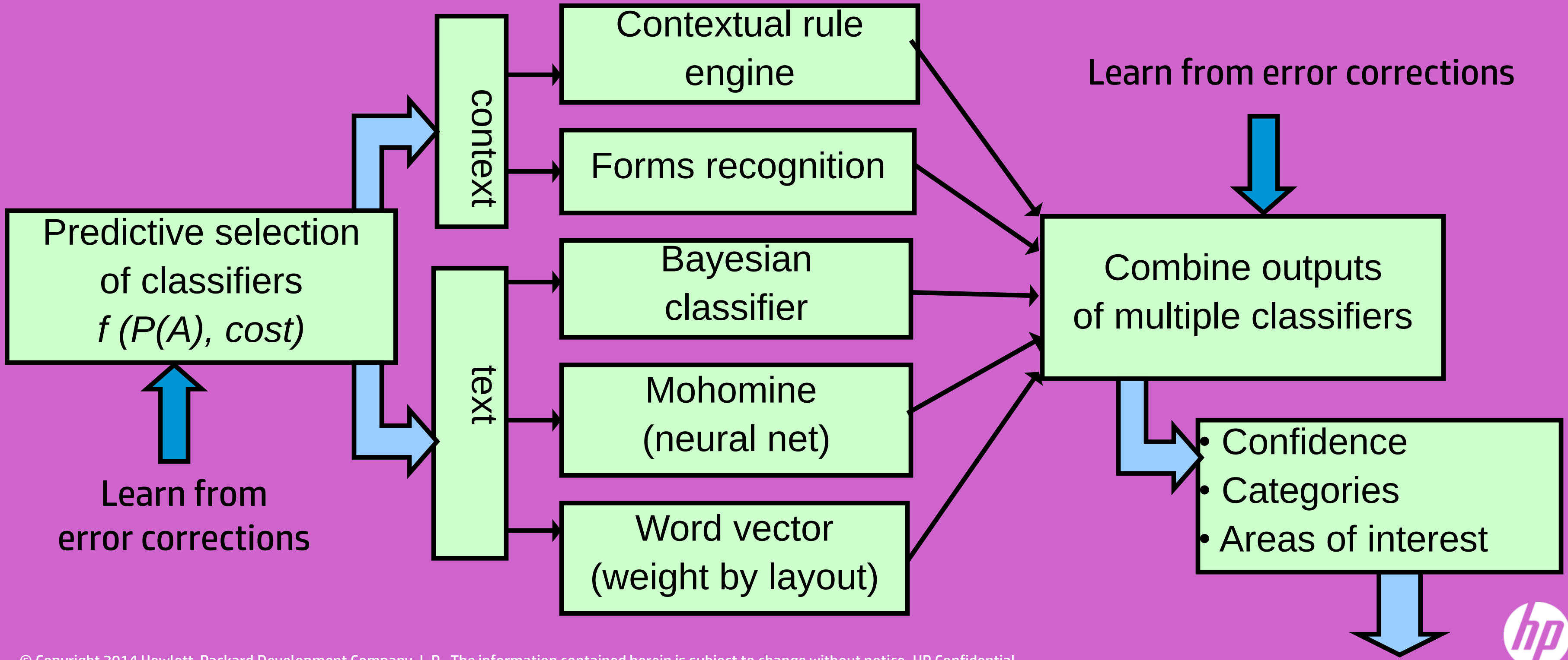
Search diversity “by priority”

The Future

There are many types of query expansion currently used. In addition to the synonymic means overviewed here, query expansion can be attained through *query augmentation*; e.g. when a query has additional terms added to it, for example other keywords associated with documents that best match the query.

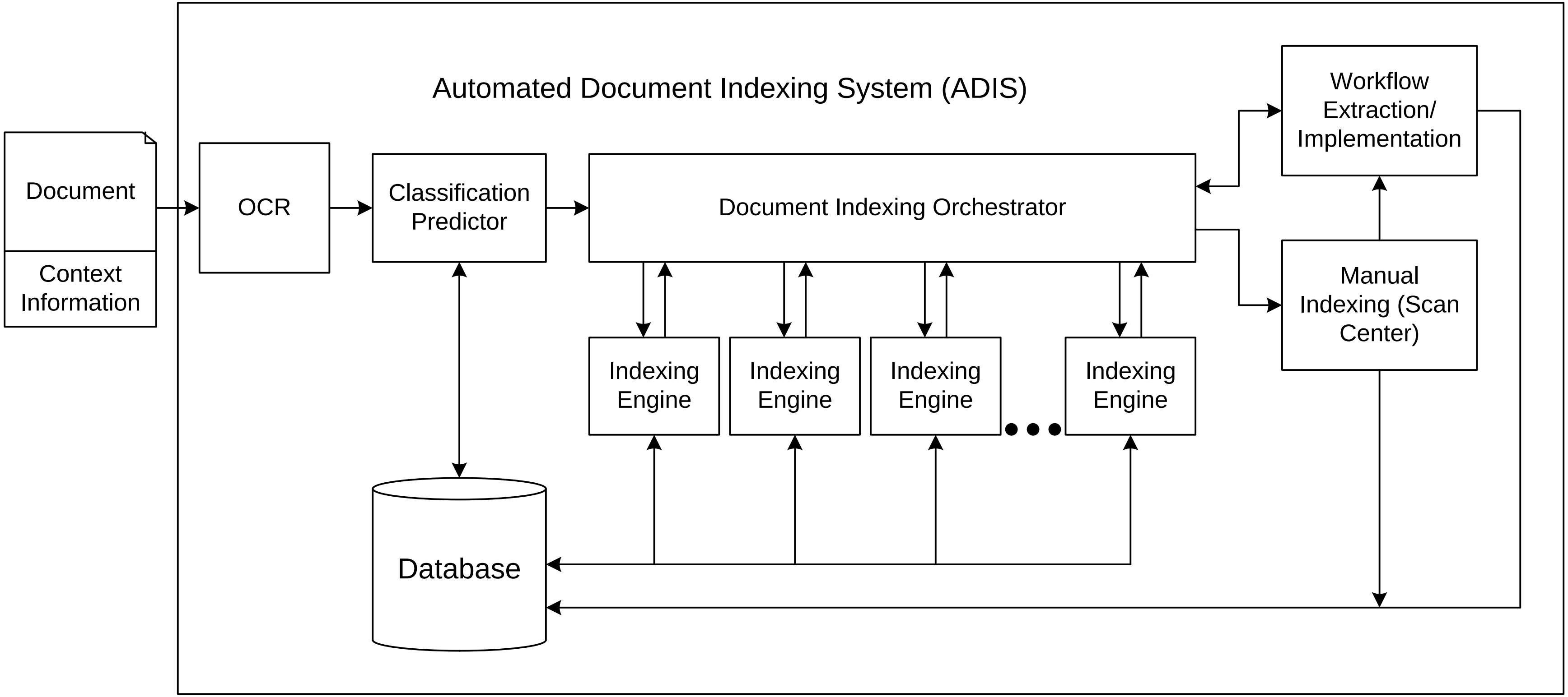
There is a logical link between these two fields of text data filtering and translation. Namely, the accuracy of the language translation approach can be directly gauged by comparing the search results on the un-translated and subsequently translated corpora. If the translation is accurate, then the documents should respond very similarly to un-translated and translated queries against the corpora. This type of functional testing of un-translated and translated corpora also warrants further, quantitative investigation.

Document Classification



Document Classification

In the Automated Document Indexing System (ADIS), up to 5 “most likely” classifications are evaluated in an effort to extract salient indices. Moving the correct classification up into this “Top 5” is thus often as important as making it the highest ranked classification.



Classifiers

Three classifiers used with a wide variety of meta-algorithmic patterns

1. Commercial-off-the-Shelf: Mohomine text classifier (neural-net based), subsequently purchased by Kofax (ww.kofax.com) and integrated into Indicius, http://www.kofax.com/products/indicius/index.asp

2. Open Source: Bayesian classifier, “Divmod Reverend" at http://divmod.org/trac/wiki/DivmodReverend.

3. In-House: “HP1” classifier that uses basic tf*idf (term frequency divided by distribution frequency – a common NLP method) calculations to classify documents.

The set of documents used for training and testing was the “20 Newsgroups” data set collected on the UC-Irvine, Knowledge Discovery in Databases Archive (http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html), which provides automatic author-specified classification.

The classifiers were trained and tested with 2.5%, 5%, ..., 17.5% and 20% of the corpus used as training data, and the remaining 97.5%, 95%, ..., 82.5%, 80% as test data.

There were 1000 documents per newsgroup, so that training took place on 500-4000 documents, and testing on the residual 16000-19500 documents.

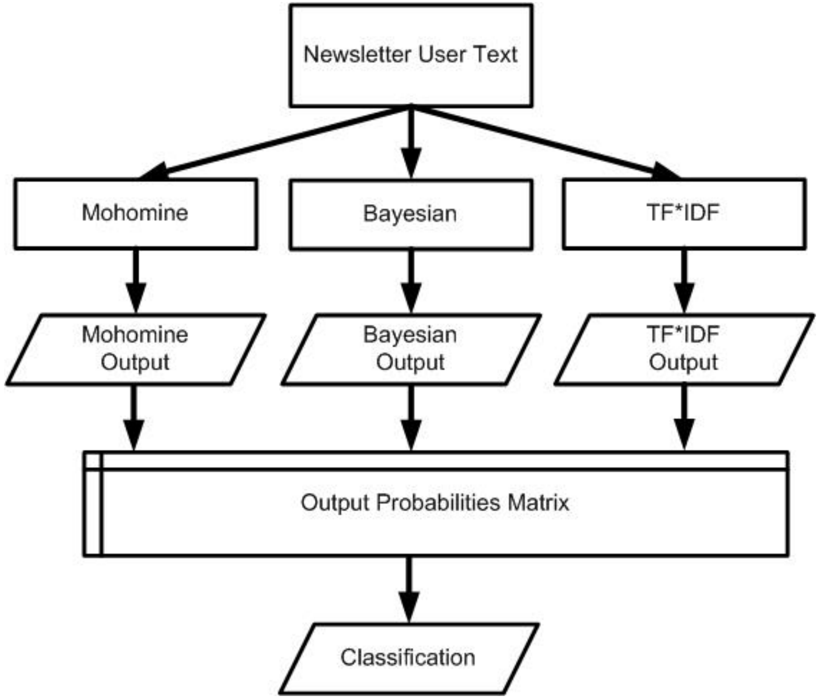
Accuracy of the three engines tested

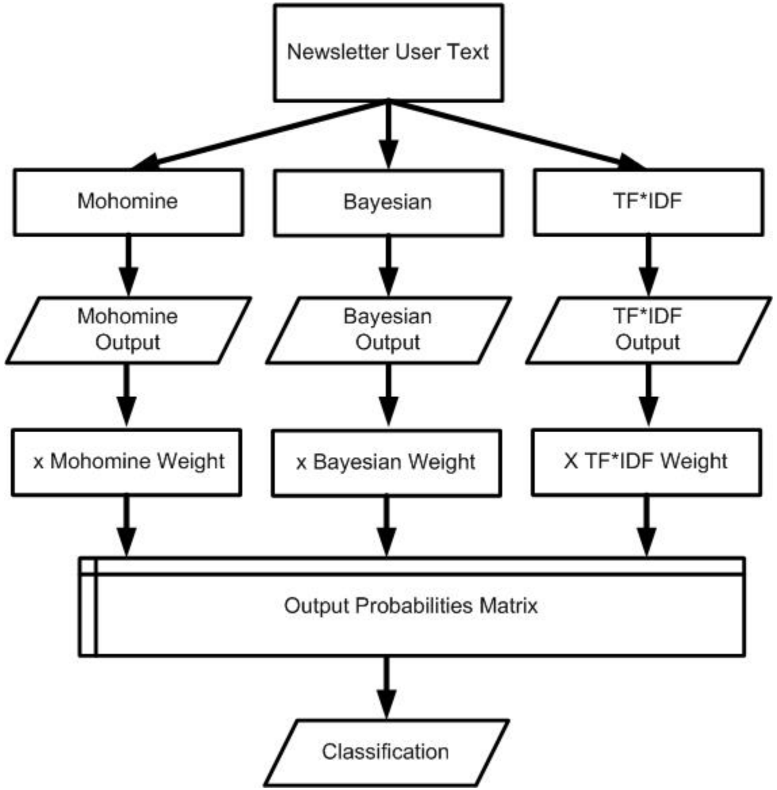
[a=2pr/(p+r)] with 2.5-20% training on the 20000 document set

#training docs (pct)	Mohomine Accuracy	Bayesian Accuracy	HP1 Accuracy
25 (2.5%)	61.1%	49.4%	47.0%
50 (5.0%)	68.1%	54.7%	56.4%
75 (7.5%)	72.1%	58.2%	61.8%
100 (10.0%)	73.4%	60.3%	64.9%
125 (12.5%)	75.1%	61.5%	67.4%
150 (15.0%)	76.4%	62.7%	69.8%
175 (17.5%)	77.4%	64.2%	71.6%
200 (20.0%)	78.2%	63.7%	72.3%

Confusion Matrix & Weighted Confusion Matrix

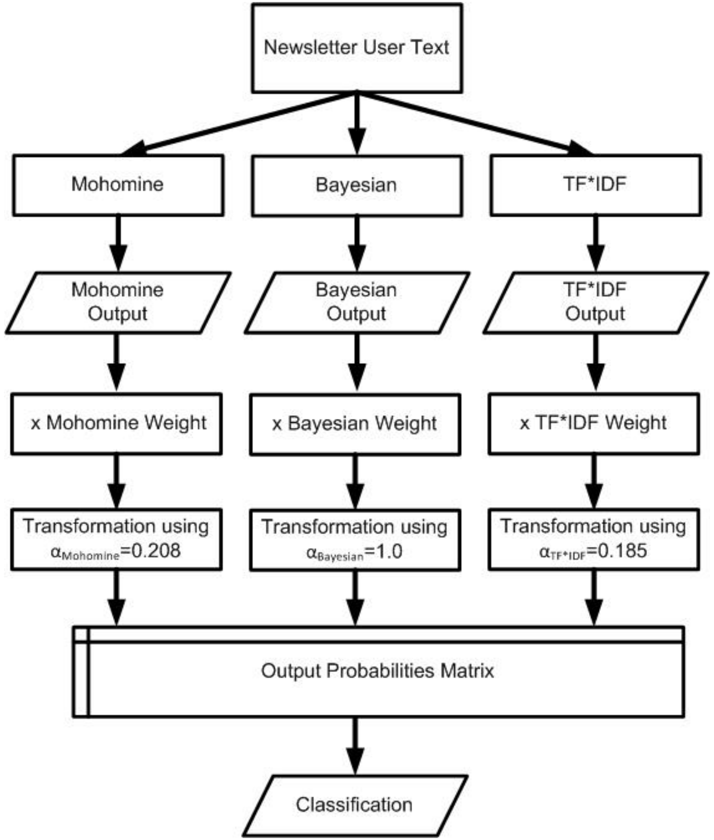
2nd Order Meta-algorithmic Patterns Applied to Text Classification

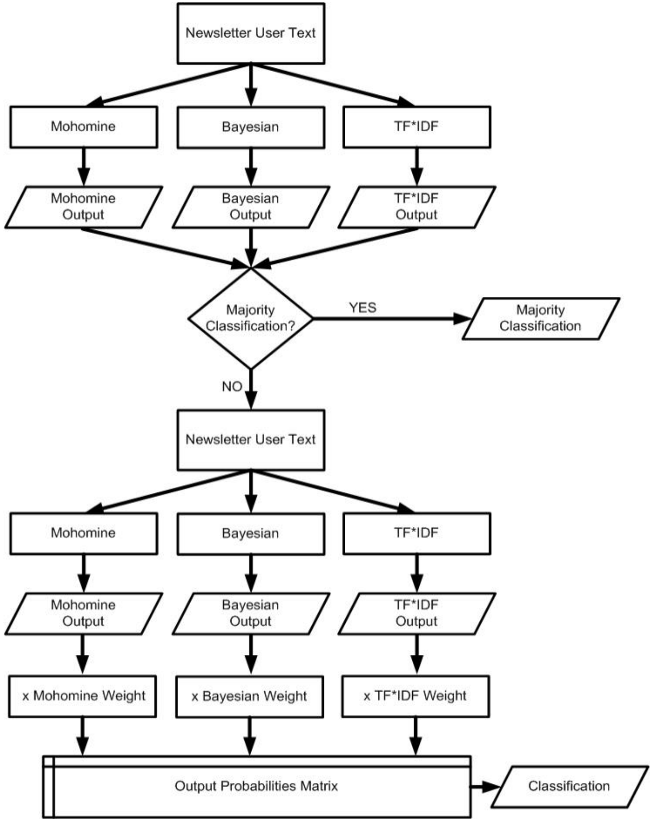




Weighted Confusion Matrix w/ Output Space Transformation & Majority Voting or Weighted Confusion Matrix Patterns

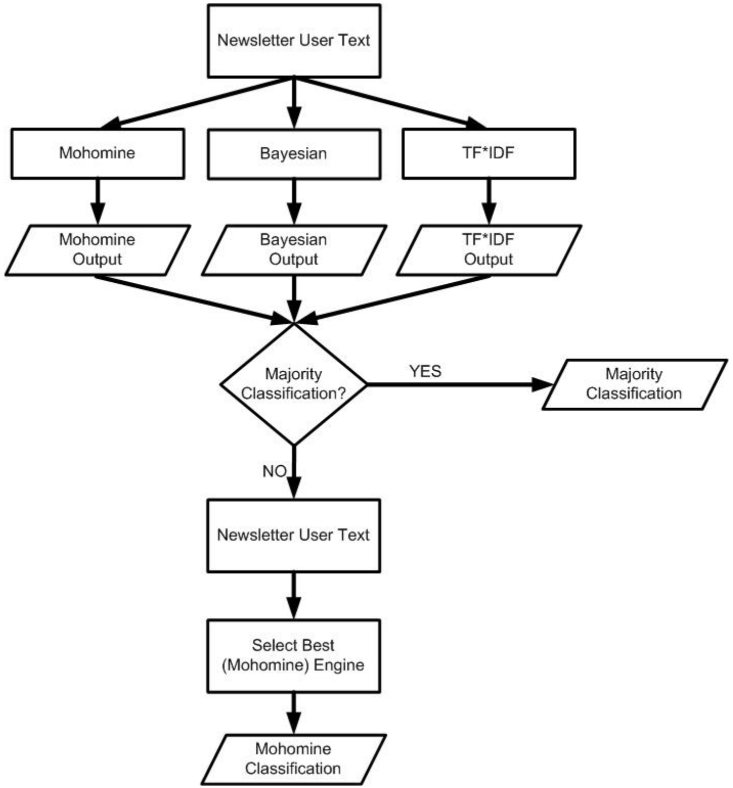
2nd Order Meta-algorithmic Patterns Applied to Text Classification

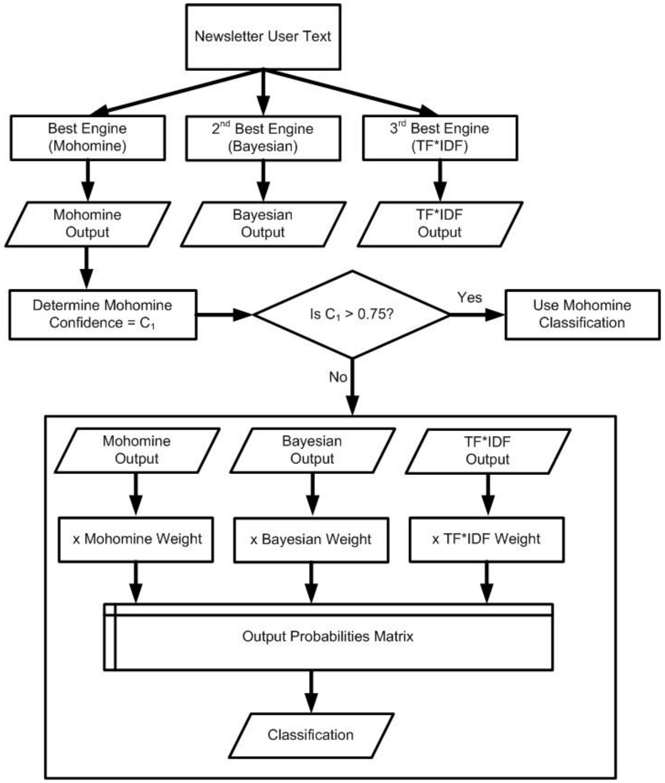




Majority Voting or Best Engine & Best Engine w/ Absolute Confidence (0.75) or Weighted Confusion Matrix Patterns

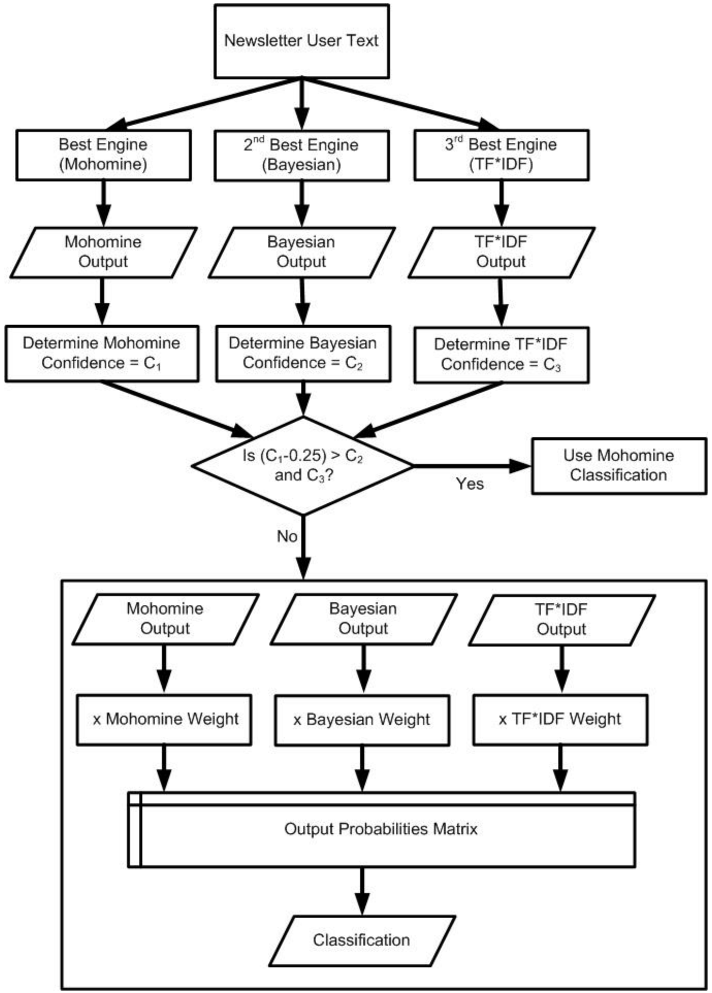
2nd Order Meta-algorithmic Patterns Applied to Text Classification





Best Engine with Differential Confidence (0.25) or Weighted Confusion Matrix Pattern

2^{nd} Order Meta-algorithmic Pattern Applied to Text Classification



Engine/Meta-Algorithmic Pattern Deployed	# Correct	# Incorrect	Accuracy (%)
Mohomine Engine	12436	3561	77.7
Bayesian Engine	9983	6014	62.4
TF*IDF Engine	11408	4589	71.3
1. Confusion Matrix and Weighted Confusion Matrix (0.56, 1.28, 0.96)	12724 12777	3273 3220	**79.5** **79.9**
2. Confusion Matrix with Output Space Transformation	12839	3158	**80.3**
3. Majority Voting or Weighted Confusion	12299	3698	76.9
4. Majority Voting or Best Engine	12265	3732	76.7
5. Best Engine with Absolute Confidence (0.75) or Weighted Confusion Matrix (threshold confidence)	12725	3272	**79.5**
6. Best Engine w/ Differential Confidence (0.25) or Weighted Confusion Matrix	12767	3230	**79.8**

Mean Rank of the Correct Classification

The Error-Reducing Effect of Multiple Engines

Approach	Mean number of classifications attempted until the correct one is obtained
100% "perfect" classifier	1.000
Bayesian	2.877
Mohomine	1.904
TF*IDF	1.891
Weighted Confusion Matrix (0.56, 1.28, 0.96)	1.533
Weighted Confusion Matrix with Output Space Transformation (1.0, 0.208, 0.185)	1.554

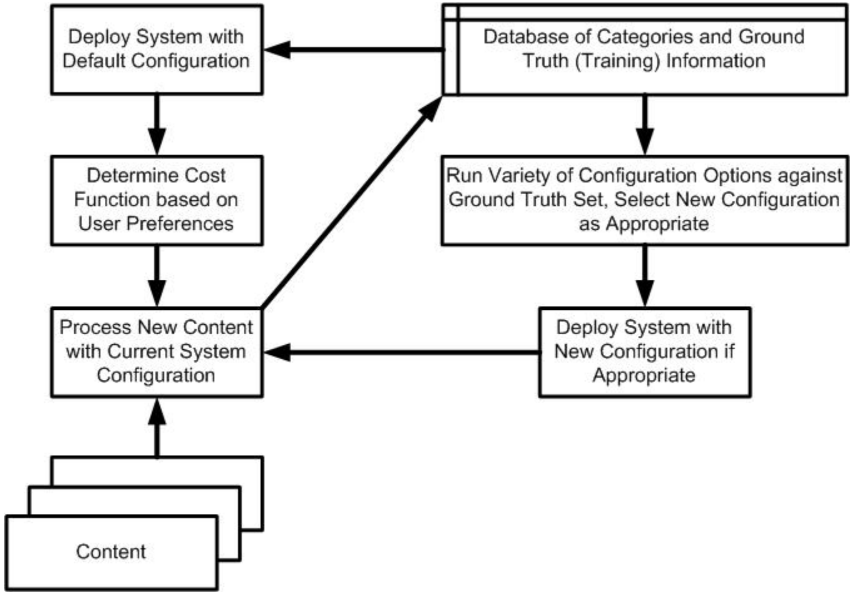
Functional Means of Optimizing Systems

Proof by Task Completion
Sensitivity Analysis
Translation

http://charliebroadway.blogspot.com/2013/03/contra-atheismum-part-5-tower-of-babel.html

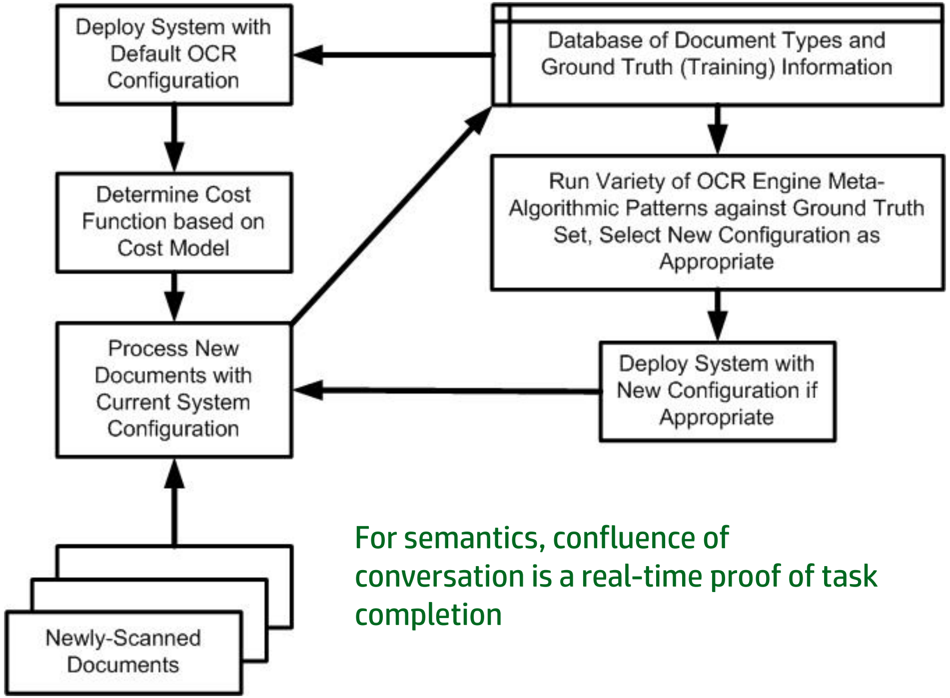
Proof by Task Completion

Generic Pattern



Dynamically changes the relative weighting of the individual generators, or the collaborative deployment of the individual generators, in response to the successful completion of intelligent tasks

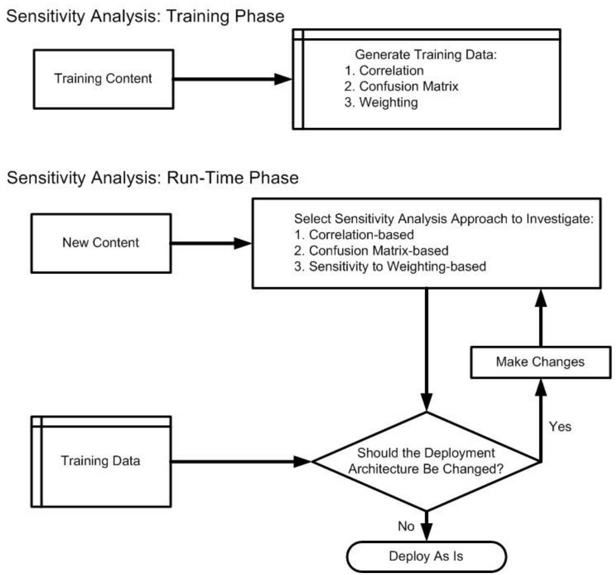
Pattern Applied to an Optical Character Recognition (OCR) Front End (Internalizes Text from "Real World")



For semantics, confluence of conversation is a real-time proof of task completion

Sensitivity Analysis

Generic Pattern



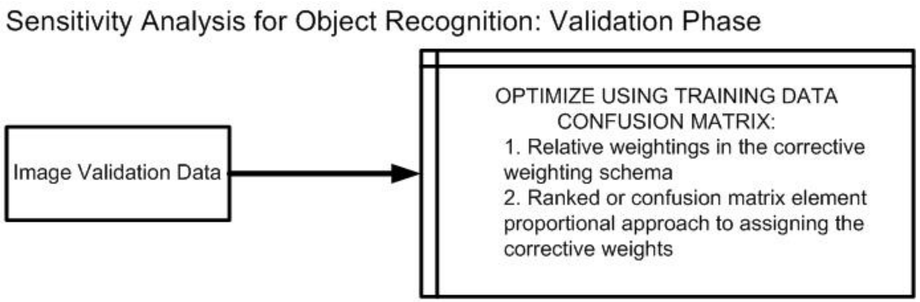
Focused on the reduction in the number of intelligence generators through:

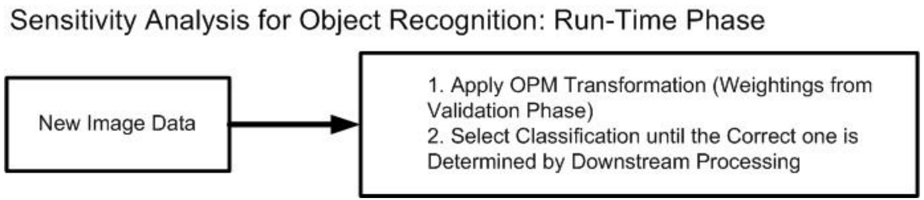
(1) correlation

(2) confusion matrix behavior

(3) selection among meta-algorithms

Pattern Applied to a Surveillance Object Recognition Example (for Media Semantics)





For semantics, confluence of conversation is a real-time proof of task completion

Other Functional Means of Assessment

Translation, Search, Classification, Categorization, Identification/Extraction

Equal Behavior

- The best translation of a set of documents results in the best match for:
  - Search behavior
  - Document classification
  - Document categorization
  - Topic identification
  - Key term/phrase extraction

  In comparing the documents in both languages

Recursive Feedback System & the Link to EM (Ouroboros)

- Translation can be optimized by finding the best match between
  - DocSet(Language 1)
  - DocSet(Language 2)

  behavior
- Search behavior itself can be optimized by using paired professionally translated documents, as can:
  - Document Classification
  - Document Categorization
  - Topic identification
  - Data mining/phrase extraction

http://www.redbubble.com/people/icefaerie/works/6929122-celtic-ouroboros?p=sticker

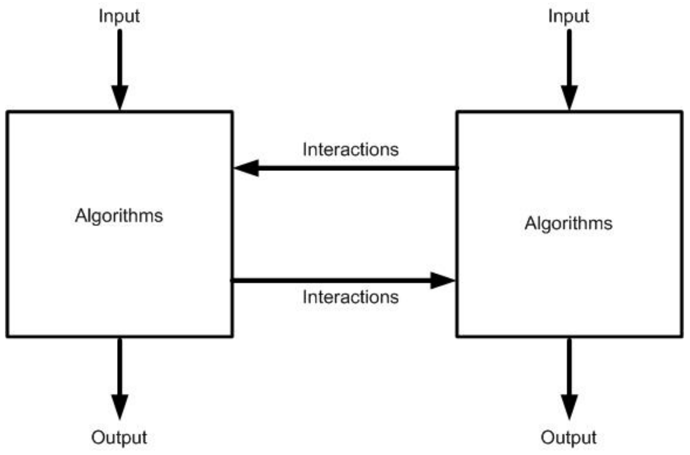
User interaction with a database

Translation, like search, is used to access desired content

Language translation is important for:

Data Mining

Knowledge Generation



Language translation can be improved through multi-engine approaches

Multiple engines can provide their output and the alignment of the output is voted on when all words are in the new language dictionary, or else the dictionary value is accepted

Language translation can be used functionally

If the translation is accurate, then the documents in each language set should respond very similarly to un-translated and translated *queries* against the corpora (equivalent translation behavior)

Language Translation

Using the Meta-Algorithmic Pattern of "Expert Feedback"

The "expert" was an English language dictionary and the sources to be translated were either in Italian and Russian.

Two 500-word documents were hand ground-truthed by the authors and three translation services were deployed.

Translator	1	2	3	Combined
Matching %	89.5	91.4	94.1	97.7

Italian-English Translation

Translator	4	5	6	Combined
Matching %	80.5	84.9	93.4	96.2

Russian-English Translation

- The words associated with the output of the multiple translations were directly aligned so that the terms could be matched directly for all three translations.
- Where the translation resulted in non-English words for one or more of the translators, the English word of another translator was used instead.
- If different English words were identified by the translations, then either the most commonly selected word or else the word provided by the engine with the greatest overall number of successes (English words provided as output) was used.

The Future

Using Figure of Speech Motifs

The revenge of semantic computing:

Semantic tools used for robust system design

Figures of Speech are one of the hardest things to understand in language

Idiomatic expressions, non-literal expressions, slang and jargon are traditionally barriers to entry for new speakers of a language

Figures of speech are therefore among the most enriching features of a language

What is the difference between Hughes and Hemingway?

“Harlem / Sent him home / in a long box- / Too dead / To know why”

It’s in what is said and what is not said

Part of speech tagging won’t help positively identify this.

Figures of Speech for System Design

Litotes

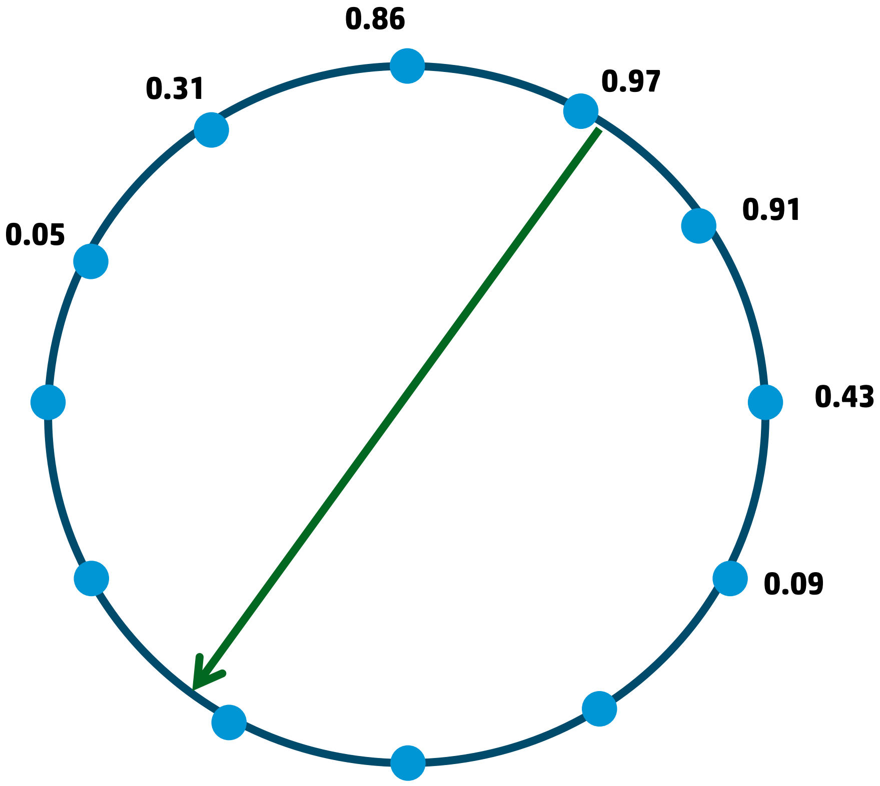
Sometimes the hardest task is finding the right answer

In classification, categorization, extraction, summarization, data mining, identification and other machine learning tasks, sometimes the hardest thing to do is find the right answer with a high degree of confidence, especially

Sometimes the easiest task is finding the wrong answer

Find the confidence (probability) that something is wrong and the vector sum points to it opposite:

Affirmation through contradiction of its opposite!



Figures of Speech for System Design

Metonymy

Metonymy-based expansion

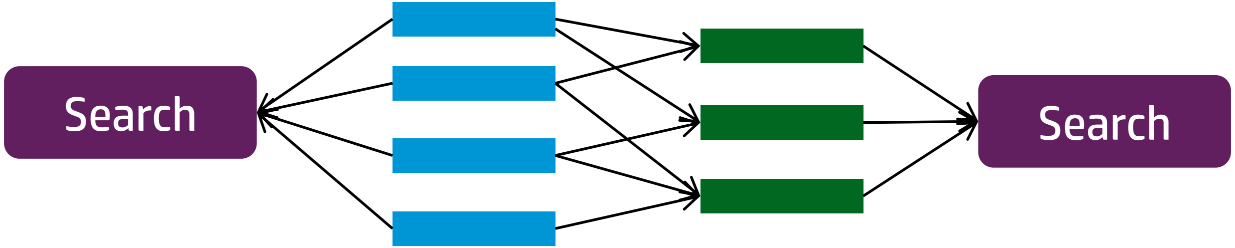
Sequentially substitute name, term, expression for closely related name, term, expression and perform search, classification, etc.

Anti-metonymy based compression

Term compression through mapping of related terms to a single concept

Hybrid system

Term compression and expansion with multiple mappings (the same terms can be mapped to multiple metonyms)



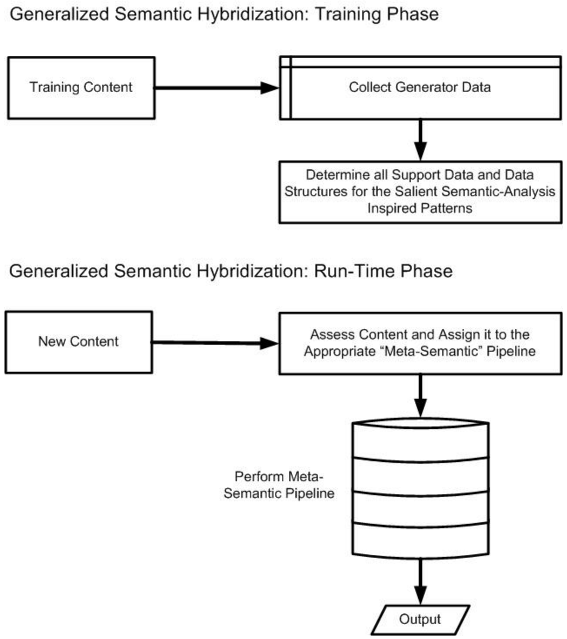
http://www.gamingfreepress.com/bermuda-triangle-discovered-in-utah-arcade/#

Figures of Speech for System Design

A New System Design "Metaphor"

Semantic analysis approaches lead to robust system design approaches

- Figure of speech building blocks
- Levenshtein and Damerau-Levenshtein operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition
- System design "poets" can weave together figure of speech based components to build more robust, more accurate, more cost-sensitive and better-performing systems



http://weblogs.baltimoresun.com/news/technology/2010/05/an_online_sarcasm_detector_yea.html

Thanks

Putting together this talk was no small task…