Few algorithms for ascertaining merit of documents and their applications

Ka.Shrinivaasan
Chennai Mathematical Institute (CMI), Chennai, India
(shrinivas@cmi.ac.in)

advised by
Dr.B.Ravindran, Indian Institute of Technology (IIT), Chennai
and
Dr.Madhavan Mukund, CMI
(ravi@cse.iitm.ac.in, madhavan@cmi.ac.in)

March 8, 2018

Abstract

Existing models for ranking documents(mostly in world wide web) are prestige based. In this article, three algorithms to objectively judge the merit of a document are proposed - 1) Citation graph maxflow 2) Recursive Gloss Overlap based intrinsic merit scoring and 3) Interview algorithm. A short discussion on generic judgement and its mathematical treatment is presented in introduction to motivate these algorithms.

1 Introduction and Motivation

Motivation for objective, independent judgement of a document is founded on the following example:

> Judge X decides about the merit of an entity Z purely by what other entities opine about Z without interacting with Z; Judge Y decides about the merit of Z by interacting only with Z. Question now is who is better judge - X or Y.

Probability of judgmental error of judge X is equal to probability of collective error of entities opining about Z while probability of judgemental error of judge Y is 0.5 as the following elementary arithmetic shows. Let us assume there are $2n$ voters and they need to decide/vote on whether a candidate is good or bad. A candidate getting majority ($n+1$ good votes) will be winner.

> Question: What is the probability that people have made a good decision?
>
> Answer: Probability of each voter making a good decision is p and bad decision is $1-p$ (0 <= p <= 1). Let p = 0.5 for an unbiased voter.

So for a candidate to be judged 'good', atleast $n+1$ people should have made a good decision. Probability of a good choice for these $2n$ voters, skipping the calculations, is :

$$P(good) = ((2n)!/4^n) * ((1/((n+1)!(n-1)!)+ \\ 1/((n+2)!(n-2)!) \\ + ... + 1/((n+n)!(n-n)!))) \quad (1)$$

If there is an objective judgement without voting, probability of good decision is 0.5. It is interesting to see that above series tends to 0.5 as n grows infinitely. Thus, the judgement-through-majority-vote error probability is equal to the error probability of judge X who uses only the inputs from witnesses to judge Z while judgement-through-interaction(without election) error probability is equal to the error probability of judge Y (i.e. 0.5) who does not use witnesses. Thus, both judges X and Y are equally fallible but the cost incurred in a real world scenario for simulating X far outweighs that of Y.Thus it is worth delving into schemes for objective judgement like Y.

2 Three algorithms presented hereunder

1. Maxflow and Path lengths of Citation graphs - objective judgement (differs from Pagerank since it is Maxflow based and not prestige based)
2. Generalized Recursive Gloss Overlap - objective judgement (simulates judge Y with a 'white-box', invasive, intrinsic merit scoring) - covers majority of this report
3. Interview algorithm - objective judgement (simulates judge Y; Uses questions and answers to judge a candidate - 'black-box' and less-invasive - and also incorporates intrinsic merit score obtained from either MaxFlow of Citation graph or Generalized Recursive Gloss Overlap)

3 Directed Graph of Citations

3.1 Average Maxflow and Path lengths of Directed Graph of Citations

Given a corpus, algorithm constructs directed graph of incoming links to a document x from those documents chronologically later than x.Thus corpus is partitioned into set of digraphs. Indegree of a vertex in this digraph reflects the importance of a document represented by a vertex. This digraph can be thought of as a flow network where concept flows from a document to others which cite. Each edge has a weight. Flow/weight for an (u,v) edge is defined as number of references v makes while citing u though there could be other ways to weight an edge. Assigning polarity to this weight is discussed in 4.2. Mincut of the digraph is the set of documents which are "potentially most influenced by the source document" (because maximum flow of concept from source occurs through this set to outside world/sink). Thus size of maxflow/mincut, averaged over all vertex-pairwise maxflow values, is a measure of influence of a source document in a community and thus points to its merit. (E.g., Chronology for web documents can be found by 'Last-modified' HTTP header which every dynamic document server is mandated to send to client). Alternative way to get the merit is to count the number of vertices in a predefined radius from source (i.e set of paths of some fixed length from source) which can be less accurate and sometimes misleading. Thus documents can be ranked using average Maxflow values. Advantage of this scheme is that it quantifies the extent of percolation of a concept within a community through Maxflow, without giving importance to the prestige measure of the vertices(documents) involved. So, this is one way of objectively assessing the merit of a vertex(document). Implementation applies Ford-Fulkerson algorithm to each s, t distinct pair and finds the average maxflow out of each vertex.

3.2 Polarity of citation edge

Parse the document/sentence containing the citation/link into tokens and find polarity. Whether a word is positive or negative can be decided by:

1. looking up a sentiment annotated ontology (e.g positivity/negativity of a lemma in SentiWordNet) or

2. entropy analysis - using $\sum_{i=0}^{1}(-P(i)logP(i))$ where P(0) = percentage of positive words and P(1) = percentage of negative words. Closer the entropy to zero, clearer the sentence/document on its viewpoint (very good or very bad) or

3. recursive gloss overlap algorithm to the citing document to get the polarity/sentiment of context citing the document.

Implementation tries all the three above. If the polarity/sentiment is negative, the weight for edge (u,v) is made negative in citation digraph, indicating a negative flow of concept to vertex v from the cited vertex u.

4 Definition Graph Convergence(or)Generalized recursive gloss overlap

4.1 Motivation for computing Intrinsic Merit of a document

Intrinsic merit is defined as the amount of intellectual effort put forth by the reader of a document and we try to quantize this effort. It is important to note that this quantized effort is independent of any observer/link-graph. Any document goes through some human understanding and we try to model it through what can be called Iceberg/Convergence/Generalized recursive gloss overlap algorithm (named so because a web document contains only a tip of the knowledge a document represents and understanding the document requires deeper recursive understanding of the facts or definitions the document is home to.).For example, going through a research paper requires the understanding of the concepts which draw a logical graph in our mind. Thus time spent on grasping the concepts and hence the intrinsic merit is proportional to the size and complexity of this graph and points to its merit (which is equal to the intellectual effort of the human reader). Since WordNet is the existing model for semantic relationship, we will try to establish that a text document can be mapped to a graph which is a subgraph of WordNet and merit can be derived applying some metrics on this graph. This is the intuition behind the algorithms that follow.

4.2 Definition tree of a document

Given a document its definition tree is recursively defined as

Definition 1. *definitiontree(all keywords of document) = definitiontree(term1) definitiontree(term2) ...definitiontree(termn) where term1, term2,...termn occur in the definition of keywords of a document.*

For example, let us consider the following document which talks about Kuratowski theorem

Document1 = Every K5,K3,3-free graph is planar

This document contains key terms like "K5,K3,3-free", "graph" and "planar".Now we recursively construct the definition tree for these terms. Key terms are decided after filtering out stopwords and by computing TF-IDF and only terms above a threshold tfidf are chosen for constructing the definition graph.

definitions at level 1:

1. K5 = Complete graph of 5 vertices (key terms: graph, vertices)
2. K3,3 = graph of two sets of 3 vertices each interconnected (key terms: graph, two sets, vertices, interconnected)
3. graph = set of vertices and edges among them (key words: vertices, edges, set)
4. planar= graph embedded on a plane (key words: graph, embedded, plane)

Thus the definition tree goes deeper as each keyword/concept is dissected and understood. Given above is level-1 grasping of the document. Important thing to note is that intersection of the sets of keywords in the definition of K5, K3,3, graph and planar is not an empty set(glosses for two or more keywords overlap). For example, intersection of definitions of K5 and K3,3 is the set {graph, vertices}. Thus the overlap of the terms "graph" and "vertices" in two definitions of K5 and K3,3 is an indication of deeper cohesion/interrelatedness of the terms in the document.Thus the replicated terms(represented by vertices) in the definition tree can be merged to get convergence (gloss overlap generalized to more than two glosses). Thus the definition tree is transformed into definition graph(since a vertex can have more than one parent) by merging replicated keyword vertices into 1 vertex. Synset definitions in WordNet gloss are used for getting keyword definitions in the implementation. But WordNet Gloss does not work for terms specialized for a domain(e.g gloss for "graph" does not have a synset for graph theory as part of its senses set). This requires ontologies for the class the document belongs to. Thus recursive gloss overlap algorithm is limited by WordNet in present implementation. At each level, word sense disambiguation is done by following Lesk's algorithm adapted to Generalized Recursive Gloss overlap to choose the synset definition fitting the context. It is important to note that 1) only one relation ("is in definition of") is used and 2) only keywords within the document are considered 3) gloss overlap is computed recursively at each level of understanding till required depth is reached.

4.3 Definition graph convergence and steps of Recursive Gloss Overlap algorithm

Convergence of a document is defined as the decrease in the number of unique vertices of the set of definition trees of its keywords from level k to level k+1 For example definition tree of the above document converges to {edges, vertices} after expanding the definition tree further down. Thus the above document has "edges" and "vertices" as its undercurrent. Thus the Convergence algorithm takes no labelled examples for inference. Only requirement is to have a dictionary/gloss/ontology of terms and their corresponding definitions. If a documents definition tree does not converge within a threshold called "depth" number of levels then the document is most likely less meaningful or of low merit. Thus the Convergence algorithms strikingly adapts an iceberg which has seemingly unconnected set of "tips" at the top but as we go deeper get unified. Level where this unification happens is a differentiator of merit. If while recursively expanding the definition tree, a vertex results in a child vertex which is same as some sibling of the parent then we compute and remove the intersection of keywords at present and previous level - since these common vertices have already been grasped. Accordingly, number of edges, vertices and relatedness are updated for each level. Number of vertices are adjusted for removal of common tokens, but number of edges remain same since they just point to a different vertex at that level. This process continues top-down till required depth is reached.

Steps:

1. Get the document as input
2. currentlevel = 1
3. keywordsatthislevel = {keywords from the document through tfidf filter (e.g > 0.02)}
4. While (currentlevel < depthrequired) {
    - For each keyword from keywordsatthislevel lookup the best matching definition for the keyword and add to a set of tokens in next level - requires WordSenseDisambiguation - implementation uses Lesk's algorithm
    - Remove common tokens with previous levels since they have been grasped in previous level (this is an optimization)
    - Update the number of vertices, edges and relatedness (vertices correspond to unique tokens, edges correspond to the single relation 'y is in definition of x' and relatedness is linear overlap or quadratic overlap) and Update tokensofthislevel
    - currentlevel = currentlevel + 1

    }

5. Output the Intrinsic merit score =

$$|vertices| \cdot |edges| \cdot |relatedness|/firstconvergencelevel$$

Where

- Relatedness $=NumberOfOverlaps$ (linear, also called as convergence factor) (or) Relatedness $= NumberOfOverlappingParents * NumberOfOverlaps^2$ (quadratic)
- firstconvergencelevel = level of first gloss overlap

At the end of recursive gloss overlap, nodes with high number of indegrees(parents) are indicators of the class of the document since greater the indegree, greater is the number of keywords overlapping (voting for an underlying theme).From graph theoretic view, Definition Graph constructed above is a multipartite graph since vertices can be partitioned into sets with no edges within a set and edges only across sets(without removal of common tokens between levels - which is only an optimization since by removing common tokens we redirect edges to vertices within the same set and multipartiteness is lost). Preserving multipartiteness is useful since it groups the tokens at each level of recursion into single set with edges across these sets - multipartite cliques of this multipartite graph can be analyzed to get the robustness. Moreover, this algorithm ignores grammatical structure. Reason is that principal differentiator in analyzing relative merit of two documents is the quality of content and complexity of content and both documents are equally grammatical.Quality of content is proportional to the vertices of the definition graph and complexity of the content is proportional to the relatedness and edges of definition graph. In spite of ignoring grammatical structure, the graph constructed above is context-sensitive since word sense disambiguation is done while choosing the synset matching a keyword. This way, the definition graph is a graph representation of the knowledge in the document sans the grammatical connectives.

4.4 Definition of shrink

Definition 2. *Let us define "shrink" to be the amount of decrease in the number of unique vertices between levels k and $k+1$ during convergence (gloss overlap)*

4.5 Comparison of two documents for relative merit - two examples

Document1 : Car plies on sky

Constructing definition graph for level-1 we get,

1. Car - automobile used for surface transport

2. plies - is flexible; goes on a surface; moves
3. sky - atmosphere; not on earth;

As can be readily seen there is overlap of 2 key terms at level 1 of the tree and thus there is less gloss overlap. Thus at level-1 document looks less meaningful.

Document2 : Cars and buses ply on road

Constructing definition graph for level-1 we get,

1. Car - automobile used for surface transport
2. Buses - automobile used for surface transport
3. ply - flexible; go on a surface; move
4. road - asphalted surface used for transport

All 4 keywords overlap giving surface as common token in their respective glosses. Overlap is better than Document1, since more keywords contribute to overlap.

4.6 Intrinsic merit score, Convergence factor and Relatedness

Definition 3. *Let us define Intrinsic merit I to be the product of number of vertices(V), number of edges(E) and Convergence factor(C) of the definition graph of the document.*

$$I = V * E * C \tag{2}$$

Convergence factor (C) is the difference between number of vertices in definition tree and number of vertices in definition graph (V). Number of vertices in definition tree includes overlapping vertices without coalescing them (since after coalescence we get the definition graph).Number of vertices in the definition tree $= x^d - 1$ where x is the average number of keywords per term definition and d is the depth of the definition tree of the document. Let us add 1 to this to get x^d (smoothing). Number of vertices in the definition graph = V Thus the Convergence factor C and Intrinsic merit I become,

$$C = x^d - V \tag{3}$$

$$I = V * E * (x^d - V) \tag{4}$$

Intrinsic Merit score can also be further fine-tuned by taking into account the level of definition tree at which first convergence(gloss overlap) happens, defined as firstconvergencelevel. Greater the firstconvergencelevel, more irrelevant the document "looks" (but has a deeper cohesion). Depth to which definition tree has to be grown is decided by extent of grasp needed by the reader. Thus greater the depth of definition tree, greater is the understanding.

It is obvious to see that Depth has to be greater than firstconvergencelevel so that some pattern can be mined from the document. Heuristically, we can grow the definition tree till intersection of leaves of all sub-trees of the keywords in the document is non-empty. This is the point where we can safely assume that all keywords in the document have been somehow related to one another. So, Intrinsic merit score can be improved by incorporating firstconvergencelevel denoted by f. Thus improved score is

$$I = V \cdot E \cdot (x^d - V)/f \tag{5}$$

(since merit is inversely proportional to firstconvergencelevel) .Complexity of constructing definition tree is O(x^d). Since non-unique vertices are coalesced(through gloss overlap), definition graph can be constructed in O(V) time (subexponential). Since x is the average number of children keywords per keyword, $x = E/V$. Substituting,

$$I = E * V * (E^d - V^d)/(V^d * f) \tag{6}$$

As an alternative to convergence factor, gloss relatedness score similar to the one discussed by Banerjee-Ted, but considering only one relation, number of overlapping parents and length of overlap can be used to get the interrelatedness/cohesion of the document. Replacing the convergence factor with relatedness, Intrinsic merit becomes, $I = V \cdot E \cdot Rel/f$ where Rel is the sum of relatedness scores, computed over all overlapping glosses at each convergence level and f is the level at which first gloss overlap occurs

$$Rel = \sum_{i=1}^{n} (relatedness(Level(i), keyword1, keyword2, ..., keywordn)) \tag{7}$$

This relatedness score has been generalized to overlap of more than two glosses with single relation R (R(x,y) = y is in definition of x). Function relatedness() for n-overlapping keywords is defined as,

$$\begin{multline} relatedness(Level(i), keyword1, keyword2, ... \\ , keywordn) = OverlapLengthAtLevel(i) \\ (LinearOverlap) \end{multline} \tag{8}$$

(or)

$$\begin{multline} relatedness(Level(i), keyword1, keyword2, ... \\ , keywordn) = n \cdot (OverlapLengthAtLevel(i)^2) \\ (QuadraticOverlap) \end{multline} \tag{9}$$

The relatedness score reflects the convergence since it takes into account the overlapping keywords at each level and length of the overlap. Thus first version of relatedness() function, implies the convergence factor (difference in number of vertices of definition tree and definition tree, signifying overlap) Intrinsic merit/Relatedness score can be used to rank the set of documents and display them to the user. Referring back to examples in 5.5, quadratic relatedness measure ((9) above) is a better choice than linear overlap since it is a function of both overlapping parents and the overlap length. The quadratic overlap gives greater weightage to length of overlap by squaring it while keeping the number of parents involved linear.

4.7 Intuition captured by above intrinsic merit score

The number of edges (representing relation between parent term and its definitions) increase as relationship among vertices of definition graph increases. The number of vertices(keywords) in the definition graph increases, as the knowledge represented by the document increases. The depth of the definition tree increases, as the understanding grows. Convergence factor increases as number of overlapping terms in definition graph increases. Similarly quadratic relatedness score increases with number of keywords involved in overlap and the length of overlap, thus pointing to stronger semantic relationship among the keywords. Intuitively, definition graph is WordNet(or any other ontology) projected onto the document.

4.8 Breadth/Depth first search of definition graph and why it is not a good choice for computing merit score

Since Breadth/Depth first search of graph can model human process of thinking, BFS/DFS algorithms can be applied to get the merit score. Since BFS/DFS algorithms run in O$(V + E)$ time merit score is proportional to $V + E$ - all vertices of the graph are visited in O$(V + E)$ time. But the drawback of this approach is that strength of underlying theme of the document and cohesion of keywords is not captured by this merit score. Since Intrinsic merit score obtained by Convergence reckons with depth and overlapping keywords, BFS/DFS merit score is discarded

4.9 Sentiment analysis applying Recursive gloss overlap

Recursive Gloss Overlap algorithm after few levels down the definition tree would spell out the sentiment of writer.

> Example1: "That movie was fantastic; Graphics was awesome"
> Keywords at level-1 of Definition graph construction:
>
> 1. movie - motion picture; positive
> 2. fantastic - good, excellent; positive
> 3. graphics - software technique; positive
> 4. awesome - good, great; positive

Overlapping terms are {good, positive} and large number of keywords(parents) contribute to this overlap. Thus the document is of extolling nature about some target entity. Prerequisite is a dictionary which annotates each word with the sentiment and sense of the word(Implementation uses SentiWordNet which gives positivity/negativity for each lemma). Sentiment analysis with Recursive Gloss Overlap is applied to finding the polarity of an edge in Citation graph (See 3.2). Recursive Gloss Overlap algorithm is applied to each Citation context and a definition graph is constructed. Keyword vertices with more than one indegree are then tested for positivity and negativity using SentiWordNet. If majority of these is positive then polarity for citation edge is positive, otherwise negative.

4.10 False negatives

Convergence algorithm never assigns lower merit score to a document which deserves a higher merit since a document with higher merit explains the concept with more depth/cohesion than document with lower merit. So false negatives do not exist

4.11 False positives

False positives exist since both a document and its arbitrarily jumbled version will get same merit score. This is prevented by assuming grammatically correct documents or by preprocessor which does parts of speech parsing to validate the grammatical structure of the document.

4.12 Normalization

Intrinsic merit can be compared only if the compared documents are of same class. Thus 2 documents explaining special relativity can be compared while a document on journalism can not be compared with a document on special relativity. Intrinsic Merit scores can be normalized by,

$$NormalizedIntrinsicMeritScore = Score/MaximumScore \tag{10}$$

4.13 Ordering and Relative Merit

Definition 4. *Document1 is more meritorious than document2 if*

1. *document1 has more keywords that need to be understood than those of document2,*

2. *cohesion/interrelation of the keywords in document1 is more than that of document2,*

3. *average number of keywords per definition is greater for document1 than document2,*

4. *firstconvergencelevel(level at which first gloss overlap occurs) of document1 is less than that of document2 and*

5. *depth of definition tree of document1 is greater than that of document2.*

If we want a weaker definition of the above, ranking may be a partial order(where some pairs of documents may not be comparable) than a total order.This appeals to intuition since document1 may be better in some aspects but worse in some other relative to document2

4.14 Semantic relatedness or Meaningfulness of a document

Definition 5. *A document is meaningful to a human reader if any pair of keywords in the document are within a threshold WordNet distance e.g Jiang-Conrath distance*

4.15 Formal proof of correctness of Convergence and Intrinsic Merit Score

Theorem 1. *If a document lacks merit, convergence(or gloss overlap) does not occur (Corollary: Document's merit is measured by extent of convergence)*

Proof. By "meritorious" document, we imply a document which is meaningful as per the definition of meaningfulness above(i.e. keywords in a document are separated within threshold WordNet distance metric like Jiang-Conrath distance). Let us denote R as a relation "is descendant of". If xRy then y is in (gloss)definition tree of keyword x(i.e y is descendant of x). If definition trees of keywords of the document are disjoint, then there is no y such that xRy and zRy for two keywords x and z. Let us define the relation S to be "two keywords are related". xSz iff xRy and zRy for some y. Thus we formalise cohesiveness/meaningfulness of a document in terms of definition graph. If a document is not meaningful then there exist no x and z such that xSz, which implies that for no y, xRy and zRy. Thus there exist no vertex y which is in definition tree of two key words. Thus convergence is a necessary condition for merit. The relation S implies that there exists a path between two keywords x and y in the document, through some intermediate nodes which are in the definition/gloss tree of x and y. There exists a threshold WordNet distance greater than length of this path since the length is finite and whether a document is meaningful depends on this threshold. Thus convergence(generalized gloss overlap) implies meaningfulness of a document as per the definition above. Moreover Intrinsic merit increases with number of edges and relatedness() - linear or quadratic. So with greater relatedness() and more number of vertices and edges, overlaps and number of nodes involved in overlap increase. This in turn implies that more number of paths are available amongst the keywords of the document since every overlap acts as a meeting point of two keyword definition trees. Probability that lengths of these paths are less than threshold WordNet distance is inversely proportional to firstconvergencelevel(level of first gloss overlap). Thus intrinsic merit score discussed earlier captures this notion. □

4.16 Parallelizability

Recursive gloss overlap is parallelizable by partitioning the tokens at each level and assigning each subset to different processors (Map) to get the tokens for next level. Individual results from processors are merged (Reduce) to get the final set of tokens for a level. This is repeated for all levels. MapReduce can be applied for parallelism.

5 Interview Algorithm (applying (1) and/or (2) for computing intrinsic merit)

5.1 Motivation for Interview algorithm

Here we map the real world scenario of an interview being conducted on a candidate where a panel asks questions and judges the candidate based on the quality of answers by candidate - candidate is a document and it is "interviewed" by a reference set of authorities. Each document x is interviewed/evaluated by set of reference documents which will decide on the merit of the document x. Reference set initially consists of n user chosen authorities on the subject. Interview is set of queries made by reference set on the document and evaluating the answer to the queries. If x passes the interview it is inducted into reference set. Next document will be interviewed by n+1 documents including last selected document and so on. Hierarchy of interviews can be built. For example Document x interviews documents y and z. Document y interviews w and document z interviews p. Thus we get a tree of interviews (it could be a directed acyclic graph too, if a candidate is interviewed by more than one reference, one of which itself was a candidate earlier). The interview scores can be weighted and summed bottom-up to get the merit of the root (Analogy: hierarchy in an organization).

5.2 Steps of the Interview algorithm

1. Relevance of the document to the reference set is measured by a classifier (NaiveBayesian or SVM or search engine results for a query)

2. Intrinsic merit score of the document is computed either by Recursive Gloss Overlap algorithm (measures the meaningfulness/sanity of the candidate) (or) by citation digraph

3. Reference set interviews the candidate and gets the score

4. Value addition of the candidate document is measured (what extra value candidate brings over and above reference set)

5. Candidate is inducted into reference set based on the above criteria if candidate is above a threshold.

5.3 Mathematical formulation of an interview

Interview is abstracted in terms of a set of tuples, where each tuple is of the form

$$t(i) = (question, answer, expectedanswer, score) \tag{11}$$

for question i.

$$Interview(I) = \{t(1), t(2), t(3), ..., t(n)\} \tag{12}$$

$$t(i).score = PercentageOfMatch(t(i).answer, t(i).expectedanswer) \tag{13}$$

$$if(\sum_{i=1}^{n}(t(i).score)) > referencethreshold \tag{14}$$

then induct the document into reference set.In the Information Retrieval context, a question is a query and the answer is the context within the document that matches the query. The answer returned by the document is then compared with expectedanswer. Comparison is done by Jaccard coefficient of shingles (n-grams)

$$t(i).score = |shingle(answer) \bigcap shingle(expectedanswer)| / |shingle(answer) \bigcup shingle(expectedanswer)| \tag{15}$$

1. Supervised: In supervised setting, each reference document is pre-equipped with user-decided set of queries and answers it expects. Thing to note is that a document is made a live object - it both has content and questions it intends to ask(set of search queries). Alternative way to compute t(i).score is to find out the definition graph of answer and expectedanswer and compute the difference between the two graphs(e.g edit distance). Downside of this is the assumption of pre-existence of correct answers which makes this a supervised learning.

2. Unsupervised: In the absence of reference questions and answers, questions that a document "intends" to ask can be thought of as set of queries for which the document has better answers(results). These set of queries/results (questions and answers) can be automatically obtained from a document through an unsupervised way by computing set of more likely to be important n-grams(by computing key phrases with tfidf above threshold) and the context of the n-grams in the reference documents.These n-grams/contexts can later be used as "references questions" (n-grams) and "reference answers"(contexts of the corresponding n-grams) to the candidate document. Thus we compensate for lack of reference Questions and Answers. Alternatively,

an interview can be simply considered as the percentage similarity of definition-graph(reference) and definition-graph(candidate) obtained by edit distance.

5.4 Searching for answer to a query within the document (as implemented)

If a document describing tourist places is given and the query is "What are the good places to visit in this city?", then query is parsed into key words like "good", "places", "visit" and "city" and matching contexts within the document are returned where context is the phrase of length $2n + 1$ (from $x - n$ to $x + n$ locations with location of keyword being x).

5.5 Value addition measure

Recursive gloss overlap algorithm gives the definition graph of the candidate document. To measure the value addition we can run the recursive gloss overlap algorithm on the reference set to get the definition graph of reference set and find out the difference between the two definition graphs - reference and candidate. Since value addition is defined as the value added which is not already present, extra vertices and edges present in candidate but absent in the reference set are a measure of value addition. Value addition can be measured by either edit distance(cost of transforming one graph to the other after adding/deleting vertices/edges), maximum common subgraph or difference of adjacency matrices. Implementation uses graph edit distance measure.

5.6 Update summarization through Interview algorithm (applying algorithm given in 6.2)

Given a news summary and a candidate news to be added to summary

1. Label the summary as reference set.
2. Run a classifier on summary and candidate to get the class to which both belong to (or get from search engine results on a news topic)
3. If class(summary) == class(candidate) proceed further
4. Calculate intrinsic merit score of the candidate news document through recursive gloss overlap algorithm described in (2) (or) from citation digraph described in (1)
5. Candidate news is interviewed by reference set(summary in this case)
6. Compute value addition of candidate to summary
7. Add the value added information from candidate into existing summary to get new summary (by getting cream of sentences with top sentence scores)

5.7 Application to Topic Detection, Link detection and Tracking

1. Interview algorithm and graph edit distance measure can be applied to news topic link detection(Answers the question - Does a pair of news stories discuss same topic?). Since same news item falls under multiple topics and is changing over time, topic of a news story is in a state of flux. Given a pair of news stories (n1, n2) execute interview of n2 with n1 as reference. This interview score decreases and edit distance grows as n2 becomes more irrelevant to n1. By defining a threshold for interview and edit distance scores to belong to same topic, link detection can be achieved. It is important to note that interview score and value addition score are inversely related.

2. At any point in time, compute edit distance for all possible pairs Nx, Ny in a topic (after getting their respective definition graphs) and choose Ny which has largest edit distance to others and hence an outlier and least likely to be in the topic. Thus topic detection is achieved(Answers the question - Does this story exist in correct topic?).

3. Topic tracking can be done by constructing definition graph and finding vertices with high number of indegrees. These keywords are voted high and point to the maximum likely topic of the news story (works as an unsupervised text classifier).This process has to be periodically done since topic of a story might change and thus the definition graph will change.

5.8 Implementation in Python

1. Get documents of same class/topic (from Reuters corpus or search engine query results) and their future references as input

2. Compute the intrinsic merit score for both documents:

   - applying citation digraph construction (or) applying definition graph convergence(generalized recursive gloss overlap)
   - Parse into keywords and get keywords above a threshold tf-idf
   - Perform WSD using Lesk's algorithm
   - Get glosses of matching sense through wordnet api
   - get overlaps at level i, update intrinsic merit score (either using linear or quadratic overlap)
   - repeat for sufficient number of levels defined by "depth"

3. execute interview if reference questions and answers are available (supervised) or through getting important n-grams/context described above (unsupervised) - at present restricted to 1-gram for keywords and bigrams for jaccard coefficient calculation

4. compute value addition through definition graph edit distance between reference and candidate, and get the score.

5. get percentage weighted sum of intrinsic merit, interview and value addition scores and get final score.

6. APPLY (2), (3) and (4) ABOVE TO NEWS UPDATE SUMMARIZATION: If final score is above threshold, update the news summary with candidate news and publish (sentence scoring is done by sum of tfidf scores of words in a sentence)

7. APPLY (2) TO GET INTRINSIC MERIT RANKING: Compare the scores of the two documents from (2) and rank. (Citation graph based maxflow ranking internally uses sentiment analysis using one of 1) Recursive gloss overlap 2) SentiWordNet 3) Entropy analysis of citation context)

8. APPLY (3) and (4) ABOVE TO TOPIC DETECTION AND TRACKING

6 Results

6.1 Results - Intrinsic Merit score with quadratic overlap for top 10 Google ranked documents for query 'data mining' sorted ascending

Pagerank - (Document,relatedness,vertices,edges,firstconvergencelevel) - IMScore

- 6 - ('ThesisDemo-datamining-test6.txt', 477660, 372, 576, 1) - 102349163520.0
- 10 - ('ThesisDemo-datamining-test10.txt', 139114790, 1172, 2339, 1) - 3.81356486745e+14
- 8 - ('ThesisDemo-datamining-test8.txt', 310161784, 1456, 3034, 1) - 1.37014092147e+15
- 7 - ('ThesisDemo-datamining-test7.txt', 310161784, 1456, 3034, 1) - 1.37014092147e+15
- 5 - ('ThesisDemo-datamining-test5.txt', 51304180926L, 2938, 10643, 1) - 1.60423730814e+18
- 4 - ('ThesisDemo-datamining-test4.txt', 99651694978L, 3324, 12921, 1) - 4.27998090689e+18
- 9 - ('ThesisDemo-datamining-test9.txt', 133686525217L, 3186, 13468, 1) - 5.73636152749e+18
- 3 - ('ThesisDemo-datamining-test3.txt', 354003740698L, 3901, 18039, 1) - 2.49112924394e+19
- 2 - ('ThesisDemo-datamining-test2.txt', 594730534291L, 3935, 20502, 1) - 4.79801059042e+19
- 1 - ('ThesisDemo-datamining-test1.txt', 2753901168066L, 5832, 33386, 1) - 5.36204253324e+20

6.2 Results - Intrinsic Merit score with quadratic overlap for top 10 Google ranked documents for query 'philosophy' sorted ascending

Pagerank - (Document,relatedness,vertices,edges,firstconvergencelevel) - IMScore

- 3 - ('ThesisDemo-philosophy-test3.txt', 63840296, 1110, 2165, 1) - 1.53417807332e+14

- 7 - ('ThesisDemo-philosophy-test7.txt', 456552729, 1234, 3041, 1) - 1.71325703153e+15
- 5 - ('ThesisDemo-philosophy-test5.txt', 915190280, 1428, 3651, 1) - 4.77146166914e+15
- 6 - ('ThesisDemo-philosophy-test6.txt', 1128268242, 1891, 4577, 1) - 9.76528235921e+15
- 2 - ('ThesisDemo-philosophy-test2.txt', 2739304610L, 2033, 5316, 1) - 2.96048373426e+16
- 10 - ('ThesisDemo-philosophy-test10.txt', 6630859968L, 2289, 6471, 1) - 9.82170869184e+16
- 9 - ('ThesisDemo-philosophy-test9.txt', 7675201402L, 2105, 6477, 1) - 1.04644348307e+17
- 8 - ('ThesisDemo-philosophy-test8.txt', 9692242200L, 2165, 6733, 1) - 1.41283281476e+17
- 4 - ('ThesisDemo-philosophy-test4.txt', 14535833906L, 2553, 7920, 1) - 2.93911072979e+17
- 1 - ('ThesisDemo-philosophy-test1.txt', 9611266377319L, 7552, 49449, 1) - 3.58922024377e+21

6.3 Results - Intrinsic Merit score with quadratic overlap for human (2 judges) judged documents on topic 'democracy' - sorted ascending

Human ranking - (Document,relatedness,vertices,edges,firstconvergencelevel) - IMScore

- 5,5 - ('ThesisDemo-democracy-test2.txt', 15535, 270, 406, 1) - 1702946700.0
- 4,6 - ('ThesisDemo-democracy-test6.txt', 60534, 253, 373, 1) - 5712533046.0
- 6,1 - ('ThesisDemo-democracy-test1.txt', 136281, 249, 384, 1) - 13030644096.0
- 2,2 - ('ThesisDemo-democracy-test4.txt', 245448, 358, 568, 1) - 49910378112.0
- 1,3 - ('ThesisDemo-democracy-test3.txt', 1623723, 364, 671, 1) - 396584600412.0
- 3,6 - ('ThesisDemo-democracy-test5.txt', 1167039, 485, 847, 1) - 479413786005.0

6.4 Results - Intrinsic Merit score with quadratic overlap for human (1 judge) judged documents on topic 'soap' - sorted ascending

Human ranking - (Document,relatedness,vertices,edges,firstconvergencelevel) - IMScore

- 4 - ('ThesisDemo-soap-test4.txt', 52, 212, 346, 2) - 1907152.0
- 3 - ('ThesisDemo-soap-test2.txt', 735, 113, 146, 1) - 12126030.0
- 2 - ('ThesisDemo-soap-test3.txt', 1368, 109, 152, 1) - 22665024.0
- 1 - ('ThesisDemo-soap-test1.txt', 2912, 188, 251, 1) - 137411456.0
- 5 - ('ThesisDemo-soap-test5.txt', 25641, 230, 353, 1) - 2081792790.0

6.5 Results - Intrinsic Merit score with quadratic overlap for top 7 Google news stories for query 'haiti earthquake' - sorted ascending

Pagerank - (Document,relatedness,vertices,edges,firstconvergencelevel) - IMScore

- 4 - ('ThesisDemo-haiti-test4.txt', 11683630, 710, 1343, 1) - 1.11406917139e+13
- 2 - ('ThesisDemo-haiti-test2.txt', 65287245, 1002, 2008, 1) - 1.31358981536e+14
- 7 - ('ThesisDemo-haiti-test7.txt', 219493417, 1258, 2785, 1) - 7.69001771262e+14
- 6 - ('ThesisDemo-haiti-test6.txt', 491851745, 1321, 3223, 1) - 2.09409962803e+15
- 3 - ('ThesisDemo-haiti-test3.txt', 4268535180L, 1966, 5693, 1) - 4.7775315353e+16
- 5 - ('ThesisDemo-haiti-test5.txt', 7043167094L, 2120, 6412, 1) - 9.57408693023e+16
- 1 - ('ThesisDemo-haiti-test1.txt', 44329850203L, 3052, 10603, 1) - 1.434529734e+18

6.6 Results - Intrinsic Merit score with quadratic overlap for top 10 Google ranked documents for query 'literary' sorted ascending

Pagerank - (Document,relatedness,vertices,edges,firstconvergencelevel) - IMScore

- 5 - ('ThesisDemo-literary-test5.txt', 38252283, 1032, 1944, 1) - 7.67420361729e+13
- 7 - ('ThesisDemo-literary-test7.txt', 815611695, 2020, 4386, 1) - 7.22609124643e+15
- 3 - ('ThesisDemo-literary-test3.txt', 5989035631L, 2039, 5938, 1) - 7.25127400033e+16
- 10 - ('ThesisDemo-literary-test10.txt', 6155411625L, 2467, 6713, 1) - 1.01939593415e+17
- 8 - ('ThesisDemo-literary-test8.txt', 296376674293L, 4598, 18333, 1) - 2.4983111474e+19
- 4 - ('ThesisDemo-literary-test4.txt', 529359994275L, 5074, 21758, 1) - 5.84413920691e+19
- 2 - ('ThesisDemo-literary-test2.txt', 643163471944L, 4920, 22433, 1) - 7.09861839373e+19
- 1 - ('ThesisDemo-literary-test1.txt', 1149789557857L, 5126, 25916, 1) - 1.52744272126e+20
- 9 - ('ThesisDemo-literary-test9.txt', 2149531315027L, 6056, 31756, 1) - 4.13385687561e+20
- 6 - ('ThesisDemo-literary-test6.txt', 3388627800057L, 6826, 36617, 1) - 8.4697952824e+20

6.7 Results Excerpt- Intrinsic Merit Ranking of Reuters corpus in 'earn' category

(Document,relatedness,vertices,edges,firstconvergencelevel) - IMScore

- ('test/15046', 34, 59, 93, 1) - 186558.0
- ('test/14911', 55, 164, 219, 1) - 1975380.0
- ('test/15213', 65, 169, 234, 1) - 2570490.0
- ('test/15063', 105, 226, 331, 2) - 3927315.0
- ('test/14899', 79, 199, 278, 1) - 4370438.0

- ('test/15070', 107, 199, 306, 1) - 6515658.0
- ('test/15185', 117, 210, 327, 1) - 8034390.0
- ('test/15074', 116, 219, 335, 1) - 8510340.0
- ('test/15103', 107, 259, 366, 1) - 10142958.0
- ('test/14965', 125, 232, 357, 1) - 10353000.0

6.8 Results - Spearman ranking coefficient and Pearson coefficient for the above rankings

- (1) . Spearman ranking coeffiecient for Google ranking for 'data mining' : 0.733333333333
- (2) . Pearson ranking coeffiecient for Google ranking for 'data mining' : 0.00888888888889
- (1) . Spearman ranking coeffiecient for Google ranking for 'literary' : 0.0909090909091
- (2) . Pearson ranking coeffiecient for Google ranking for 'literary' : 0.00110192837466
- (1) . Spearman ranking coeffiecient for Google ranking for 'philosophy' : 0.0424242424242
- (2) . Pearson ranking coeffiecient for Google ranking for 'philosophy' : 0.000514233241506
- (1) . Spearman ranking coeffiecient for Google ranking for 'haiti earthquake' : 0.25
- (2) . Pearson ranking coeffiecient for Google ranking for 'haiti earthquake' : 0.00892857142857
- (1) . Spearman ranking coeffiecient for Human ranking(2 judges) for 'democracy' : 0.385714285714
- (2) . Pearson ranking coeffiecient for Human ranking(2 judges) for 'democracy' : 0.0174334140436
- (1) . Spearman ranking coeffiecient for Human ranking(1 judge) for 'soap' : 0.9
- (2) . Pearson ranking coeffiecient for Human ranking(1 judge) for 'soap' : 0.09

As per Spearman coefficient, correlations between Google ranking and Recursive Gloss Overlap are 73%, 4%, 9% and 25% while with human ranking they are 38% and 90%

6.9 Results - Update Summarization applying Interview algorithm with Recursive Gloss Overlap - Example - Summary size is 12.5%

Candidate document:

> 1 Dead in Bangkok Protests, More Than 70 Wounded
>
> VOA News22 April 2010 A Thai woman lies injured on the ground in Bangkok after several small explosions occurred near site of anti-government protests in Bangkok, 22 Apr 2010 Photo: AP
>
> A Thai woman lies injured on the ground in Bangkok after several small explosions occurred near site of anti-government protests in Bangkok, 22 Apr 2010
>
> Hospitals in Thailand's capital, Bangkok, say at least one person has been killed and many others wounded in a series of explosions near an encampment of anti-government protesters.
>
> More than 70 people were reported wounded Thursday at the camp in the city's business district, which is packed with armed troops and diverse groups of protesters. Reports say at least five hand grenades exploded, prompting the closure of a nearby train station.
>
> The protesters have besieged central Bangkok for weeks, trying the patience of citizens and business leaders. A coalition of groups gathered to drive the so-called Red Shirt protesters from the main retail and tourist district. Police separated the two sides.
>
> The Red Shirt protesters have rallied for five weeks, demanding new elections and the resignation of Prime Minister Abhisit Vejjajiva.
>
> A coalition opposed to the Red Shirts - called the Multi-Colored Shirts - has announced a mass rally for Friday.
>
> Most of the protesters support former Prime Minister Thaksin Shinawatra, who was ousted in 2006. Mr. Thaksin lives in exile and faces a prison sentences on corruption charges in Thailand. He has a significant following among the country's rural and low-income population.
>
> Mr. Abhisit came to power in December 2008, after months of massive anti-Thaksin protests by demonstrators known as the Yellow Shirts.
>
> The military said soldiers will use tear gas, rubber bullets and live ammunition, if necessary, to remove the protesters. But Army Chief General Anupong Paochinda has been reluctant to use arms fearing renewed bloodshed.
>
> An April 10 clash between the Red Shirts and Thai security forces left at least 25 people dead, and 850 others injured.

Reference document:

> Bangkok blasts kill one, injure 75 - Thai media 11:16am EDT
>
> BANGKOK, April 23 (Reuters) - A series of grenade blasts that rocked Bangkok's business district on Friday killed at least one person and wounded 75, hospitals and the Thai media said.
>
> Five M-79 grenades hit an area packed with heavily armed troops and studded with banks, office towers and hotels. Four of the wounded had serious injuries, including two foreigners, according to witnesses, hospital officials and an army spokesman. (Additional reporting by Nopporn Wong-Anan; Writing by Jason Szep; Editing by Bill Tarrant)

Summary:

> *1 Dead in Bangkok Protests, More Than 70 Wounded*
>
> *VOA News22 April 2010 A Thai woman lies injured on the ground in Bangkok after several small explosions occurred near site of anti-government protests in Bangkok, 22 Apr 2010 Photo: AP*
>
> *A Thai woman lies injured on the ground in Bangkok after several small explosions occurred near site of anti-government protests in Bangkok, 22 Apr 2010*
>
> *Hospitals in Thailand's capital, Bangkok, say at least one person has been killed and many others wounded in a series of explosions near an encampment of anti-government protesters . Bangkok blasts kill one, injure 75 - Thai media 11:16am EDT*
>
> BANGKOK, April 23 (Reuters) - A series of grenade blasts that rocked Bangkok's business district on Friday killed at least one person and wounded 75, hospitals and the Thai media said .

6.10 Results Excerpt - Topic Detection and Tracking applying Interview algorithm with Recursive Gloss Overlap

Example Reference News Story (ThesisDemo-ipad-test1) - Topic 'ipad':

> Apple unveils iAd platform; iPad sales look strong Photo 6:29am EDT
>
> By Gabriel Madway
>
> CUPERTINO, California (Reuters) - Apple CEO Steve Jobs showed off a new smartphone operating system on Thursday that features an advertising platform to compete with Google's, and revealed stronger-than-expected sales of 450,000 units for the iPad.
>
> The iPhone 4.0 software will be available on Apple's hugely popular smartphone this summer, complete with a number of

upgrades, including a long-awaited multi-tasking capability that allows the use of several applications at once.

A version of the iPhone’s operating system is also used on the iPad, and the latest generation of software will come to Apple’s new tablet computer this fall.

The new advertising platform for the iPhone and iPad – dubbed iAd – marks Apple’s first foray into a small but growing market, and is sure to please the thousands of application developers who make their living off those devices, providing them with a new revenue stream.

The iPad’s early sales impressed analysts, many of whom expect 1 million units to be sold in the quarter ending June, and roughly 5 million in 2010, though estimates vary widely.

"We’re making them as fast as we can. Our ramp is going well, but evidently we can’t quite make enough of them yet so we’re going to have to try harder," Jobs said, noting iPad sellouts at Best Buy stores.

The electronics giant has staked its reputation on the 9.7-inch touchscreen tablet, essentially a cross between a smartphone and a laptop. It is helping foster a market for tablet computers that is expected to grow to as many as 50 million units by 2014, according to analysts.

"I think it’s pretty impressive, five days almost half a million units, and it shows there’s still pretty good momentum behind the first day," said Gartner analyst Van Baker.

Despite critics who question whether a true need exists for such a gadget, analysts expect Hewlett-Packard, Dell and others to trot out their own competing devices this year.

Since the iPad went on sale on April 3, users have downloaded 600,000 digital books and 3.5 million applications for the device, Jobs said. There are already 3,500 apps available for the iPad.

"It was above my expectations, frankly," said Joe Clark, managing partner of Financial Enhancement Group, referring to iPad sales. "The day the original Apps Store launched it was a game change for the iPhone and it will do the same eventually for the iPad."

At a media event at the company’s Cupertino, California, headquarters, Jobs said Apple had so far sold more than 50 million iPhones, the smartphone that competes with Research in Motion’s Blackberry and Motorola’s Droid.

That implies that the company sold 7 million or more devices in the March quarter, which would be above many analysts’ forecasts.

MOBILE AD WAR

Apple is expected to launch the fourth-generation model of its iPhone, which was introduced in 2007, later this year.

Pancreatic cancer survivor Jobs, looking thin but energetic, introduced the iAd mobile platform, which he said had the opportunity to make 1 billion ad impressions a day on tens of millions of Apple mobile device users.

IAds will allow applications developers to use advertisements in their apps, pocketing 60 percent of the revenue. Apple will sell and host the ads.

Jobs harshly criticized the current manner and look of mobile advertising, particularly search ads. He promised that iAds will foster more engaging advertising that will not pull users away from the content within apps.

NEW ARENA

Tim Bajarin, president of consulting company Creative Strategies, said it was a dramatic shift in thinking about the delivery of mobile ads, and an obvious move by Apple to set itself apart from Google Inc, which made its name on search ads.

"It's very clear that Jobs believes that ads in the context of apps makes more sense than generic mobile search," he said.

Apple's entry into the mobile ad arena had been widely expected. This year, it paid $270 million for Quattro Wireless, an advertising network that spans both mobile websites and smartphone applications.

Google, which already sells advertising on smartphones, agreed to buy mobile ad firm AdMob late in 2009. U.S. regulators are examining the deal's antitrust implications.

Jobs said Apple was also in the hunt to buy AdMob before Google "snatched them from us because they didn't want us to have them." The comments were just the latest hint at the rift that has emerged between Apple and Google, which were once allies but now compete in a number of arenas.

Research group Gartner expects the mobile advertising market to expand by 78 percent to $1.6 billion in 2010.

Jobs also said the new operating system will include support for multi-tasking – addressing a perennial consumer complaint – allowing users to switch between several programs running simultaneously.

Shares of Apple turned positive briefly after Jobs' announcement, before quickly dipping back into negative territory. They closed 0.3 percent lower at $239.94 on the Nasdaq. (Writing by Edwin Chan; Editing by Steve Orlofsky, Leslie Gevirtz and Matthew Lewis)

Example Candidate News Story (ThesisDemo-dantewada-test1) - Topic 'Dantewada':

> Chidambaram offers to quit, PM says no CNN-IBN
>
> New Delhi: Union Home Minister P Chidambaram has reportedly offered to resign taking responsibility for the Dantewada massacre in which the Maoists butchered 76 security personnel. However, Prime Minister Manmohan Singh has rejected Chidambaram's offers to resign.
>
> Chidambaram reportedly met Prime Minister Manmohan Singh taking "full responsibility" for the Dantewada attack on a CRPF patrol party and offered to step down.
>
> Sources say that a section within the Congress party has been unhappy with the way Chidambaram has handled the Maoist issue so far including his tough talk on the rebels and his friction with West Bengal Chief Minister Buddhadeb Bhattacharjee.
>
> When the Home Minister came back from his Dantewada tour he reportedly met the Prime Minister and told him that if he (Prime Minister) was dissatisfied with his performance, he was ready to step down.
>
> Earlier, on Friday while attending the Valour Day function of the CRPF, Chidambaram said, "I salute the CRPF. I promise that government will always stand by you. Where does the buck stop after Dantewada? The buck stops at my desk. I accept full responsibility of what happened in Dantewada. I told this to the Prime Minister as well."
>
> In the past whenever there was talk of all-out action to control Maoists, it was usually tempered by the Congress party that the issue it was a socio-economic one and had to be dealt with in such a manner.
>
> Many leaders had been maintaining that Maoists were our own people and so they should not be dealt with in the same firm way as one would deal with terrorists.
>
> But now that stand seems to have been diluted a bit within the Congress following the massacre of the CRPF team on Tuesday.
>
> The party, too, has begun to realise that to go soft in the weeding out Maoists will not really go down very well.
>
> So Chidambaram's offer to step down accepting complete responsibility is seen as a political masterstroke by him and also a plus point in his favour.
>
> A large group of Maoists had ambushed the CPRF team belonging to the 62 Battalion between 6 AM and 7 AM on Tuesday between Chintalnar and Tademetla villages in Sukma block of Dantewada

> district of Chhattigarh when the security personnel were on the way to Tademetla in a vehicle.
>
> The Maoists, believed to be between 200-1000, first triggered a land mine destroying the vehicle carrying the security personnel. In the ensuing gunbattle 75 CRPF men and one local police constable were killed.

Topic Link Detection(to check if above two news stories discuss same topic):

- Interview score:3.09090909091
- Interview score(in percentage correctness): 10.303030303
- Edit Distance (as percentage value addition from reference):79.4014084507
- Topic Link Detection - ThesisDemo-ipad-test1.txt and ThesisDemo-dantewada-test1.txt do not discuss same topic

Topic Detection(to check if a news story is under correct topic) - Topics are ’ipad’(1 story), ’sukhna’(2 stories), ’lufthansa’(1 story),’dantewada’(1 story):

- Topic Detection - News story ThesisDemo-ipad-test1.txt has largest pairwise editdistance from ThesisDemo-sukhna-test1.txt and least likely to be in this topic
- Topic Detection - News story ThesisDemo-ipad-test1.txt has largest pairwise editdistance from ThesisDemo-lufthansa-test1.txt and least likely to be in this topic
- Topic Detection - News story ThesisDemo-ipad-test1.txt has largest pairwise editdistance from ThesisDemo-sukhna-test2.txt and least likely to be in this topic
- Topic Detection - News story ThesisDemo-ipad-test1.txt has largest pairwise editdistance from ThesisDemo-dantewada-test1.txt and least likely to be in this topic
- Topic Detection - News story ThesisDemo-dantewada-test1.txt has largest pairwise editdistance from ThesisDemo-ipad-test1.txt and least likely to be in this topic

6.11 Results Excerpt - Applying Sentiment Analysis with Recursive Gloss Overlap for finding polarity of an edge in Citation Graph Maxflow (1)

- Nodes with more than 1 parent (and hence the most likely classes of document) are: set([’good’, ’used’])
- getPositivity: good

- getNegativity: good
- getPositivity: used
- getNegativity: used
- negative words: []
- positive words: ['good', 'used']

6.12 Results Excerpt - Citation graph maxflow with a simple link graph example - polarity determined by Recursive Gloss Overlap

- Following is the adjacency matrix for hyperlink graph amongst 7 html documents (file1.html, file2.html, file3.html, file4.html, file5.html, file6.html, file7.html)

  $$Adjacency \quad Matrix \quad of \quad Link \quad Graph = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ -1 & 1 & 1 & 0 & -1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Average concept maxflow out of each page:
  - 'file2.html': 3.7142857142857144
  - 'file4.html': 3.2857142857142856
  - 'file5.html': 2.0
  - 'file3.html': 3.4285714285714284
  - 'file7.html': 0.0
  - 'file1.html': 3.4285714285714284
  - 'file6.html': 0.0
- Number of nodes within radius 2 of the source file2.html = 7
- Number of nodes within radius 2 of the source file3.html = 7
- Number of nodes within radius 2 of the source file4.html = 7
- Number of nodes within radius 2 of the source file5.html = 6
- Number of nodes within radius 2 of the source file1.html = 7
- Number of nodes within radius 2 of the source file6.html = 0

- Number of nodes within radius 2 of the source file7.html = 0

Average maxflow values above show that file2, file1, file3, file4, file5, file6, file7 are ranked in descending order of their Maxflow merit.Thus file2 is most influential in this community. The radius measure with length 2 gives the ranking: file2, file3, file4, file1 all tied in first and file5 ranked next.

6.13 Conclusion

Motivation for this excercise, as evident from results above, is to explore the possibility of finding an algorithm/framework to assess the merit of a document with and without link graph structure in place with greater emphasis on the latter. Citation graph maxflow measures the penetration of a concept (represented in a document), in a link graph while the Recursive gloss overlap objectively judges the document without getting inputs from any incoming links. Interview algorithm uses either of these two algorithms and abstracts some real world applications.As seen above Google rankings differ (with some exceptions) from Recursive gloss overlap intrinsic merit rankings which is on the expected lines - content-and-complexity based merit scoring is not necessarily same as popularity based ranking. Moreover the intrinsic ranking scheme given above need not be the only possible way of computing merit. Once we have definition graph for a document (whether multipartite or not), multitude of more ranking schemes can be invented - for example based on 1) k-connectedness of the definition graph 2) completeness or robustness of the definition graph 3) (multipartite) cliques of (multipartite) definition graph (if multipartite) etc., Since definition graph construction is computationally intensive, there is a scope of improvement in improving the recursive gloss overlap algorithm by applying some parallel processing framework like MapReduce. Applying Evocation WordNet, implementing a MapReduce(e.g Hadoop) cluster and considering more than one relation are future directions to think about. Theoretical foundation for the recursive gloss overlap comes from WordNet itself which visualises the relatedness of words - Definition graph is just an induced subgraph of WordNet for a document. Since merit quantification of a document can not be done without analyzing relatedness of keywords in a document, definition graph, which is a subgraph of WordNet for a document, is a plausible representation. Accuracy of Recursive gloss overlap depends on the accuracy of WordNet, depth to which definition trees are grown and Word Sense Disambiguation.

References

[1] Graph Similarity, Master's thesis by Laura Zager and George Verghese EECS MIT 2005

[2] Edit distance and its computation, presentation by Joszef Balogh and Ryan Martin

[3] Extended Gloss Overlaps as a measure of semantic relatedness, Satanjeev Banerjee and Ted Pedersen

[4] Sematic Language Models for TDT, Ramesh Nallapati,University of Amherst

[5] WordNet Evocation Project-
http://wordnet.cs.princeton.edu/downloads/evocation/release-0.4/README.TXT

[6] SentiWordNet - http://sentiwordnet.isti.cnr.it/

[7] WordNet - http://wordnet.princeton.edu

[8] Navigli, R. 2009. Word sense disambiguation: A survey. ACM Comput. Surv. 41, 2, Article 10 (February 2009)

[9] Temporal information in Topic Detection and Tracking - Juha Makkonen, University of Helsinki

[10] Overview NIST Topic Detection and Tracking by G. Doddington ,
http://www.itl.nist.gov/iaui/894.01/tests/tdt/tdt99/
presentations/index.htm

[11] Topic Detection and Tracking Pilot Study - James Allan , Jaime Carbonell , George Doddington , Jonathan Yamron , and Yiming Yang UMass Amherst, CMU, DARPA, Dragon Systems, and CMU

[12] The cognitive revolution: a historical perspective , George A. Miller Department of Psychology, Princeton University, TRENDS in Cognitive Sciences Vol.7 No.3 March 2003

[13] On Bipartite and Multipartite Clique Problems, Milind Dawandeb, Pinar Keskinocakc, Jayashankar M. Swaminathan and Sridhar Tayur,Journal of Algorithms 41, 388âĂŞ403 (2001)

[14] Python Natural Language Toolkit - http://nltk.sourceforge.net

[15] Partitioning CiteSeer's Citation Graph - Revised Version , Gregory Mermoud, Marc A. Schaub, and Gregory Theoduloz, School of Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne (EPFL), 1015 Lausanne, Switzerland

[16] Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, Alexander Budanitsky and Graeme Hirst, Department of Computer Science, University of Toronto

[17] The Official Python Tutorial - http://docs.python.org/tut/tut.html

[18] MapReduce: Simplified Data Processing On Large Clusters, Jeffrey Dean and Sanjay Ghemawat, Google Inc.,

[19] Opinion Mining and Summarization - Sentiment Analysis , Bing Liu Department of Computer Science , University of Illinois at Chicago , Tutorial given at WWW-2008, April 21, 2008 in Beijing WWW 2008

[20] Web Data Mining, Bing Liu, Department of Computer Science , University of Illinois at Chicago

[21] Introduction to Algorithms - Second Edition, Thomas H.Cormen, Charles E.Liecerson, Ronald L.Rivest, Clifford Stein

Decidability of Existence and Construction of a Complement of a given function

Ka.Shrinivaasan, Chennai Mathematical Institute (CMI)
(shrinivas@cmi.ac.in)

April 28, 2011

Abstract

This article defines a complement of a function and conditions for existence of such a complement function and presents few algorithms to construct a complement.

1 Goal

Goal is to analyze decidability of finding a complement of a function defined over both domain and range as integers. There are two cases to consider:

1. The domain and range of the function are infinite
2. The domain and range of the function are finite

2 Example of a complement function (when the domain and range are infinite)

Consider an infinite set of positive integers = $\{1, 2, 3, 4, 5...\}$ .Let us say we create a subset of even integers out of it = $\{2, 4, 6, 8, ...\}$. This set of even numbers can be generated by using the function $f(x) = 2x$, x=0,1,2,3... A complement of this set is infinite set of odd numbers and they can be generated using g(x) = 2x+1 , x=0,1,2,3,4,... So we define $g(x) = 2x + 1$ to be *a complement function* of $f(x) = 2x$ over the set of integers.
Similarly for the function $f(x, y) = xy$, a complement function (ignoring trivial factors 1 and n) is over the set of all primes. Thus finding a complement g of f(x,y) amounts to finding distribution of primes i.e g shows a pattern in primes.This was a question posted in google group sci.math ($http : //groups.google.com/group/sci.math/browse\_thread/thread /46ab0335ce9205d9/f3c67ee19926e02e$)

3 Existence of a complement when domain and range are finite

3.1 Definitions

1. Let U be a universe of n-bit integers of size 2^n integers
2. Function f_n is defined over the domain of a-bit integers and range U of n bit integers - f:$\{0,1\}^a$ - $\{0,1\}^n$ for $a > 0$ and $n > 0$
3. Function g_n is defined over the domain of b-bit integers and range U of n bit integers - g:$\{0,1\}^b$-$\{0,1\}^n$ for $b > 0$ and $n > 0$
4. $L(f_n) =$ language generated by the function $f_n = \{f(x_0), f(x_1), ..., f(x_{2^a-1})\}$
5. $L(g_n) =$ language generated by a complement function $g_n = \{g(x_0), g(x_1), ..., g(x_{2^b-1})\}$
6. Function g_n is defined as *a complement* of f_n if
   (a) g_n is total
   (b) $g_n(x) \neq f_n(x) \quad \forall x$
   (c) $L(f_n) \cup L(g_n) = \{0,1\}^n$ and
   (d) $L(f_n) \cap L(g_n) = \phi$
7. The subscript n for the functions f and g denotes the finiteness of the size of input and output.

3.2 Conditions for existence of a complement

For a complement to exist the universe U - the range of functions f_n and g_n - must be partitioned by the functions f_n and g_n. Due to this, if

1. f_n:$\{0,1\}^a$ - $\{0,1\}^n$ and
2. g_n:$\{0,1\}^b$ - $\{0,1\}^n$

then the inequality $2^a + 2^b >= 2^n$ must hold. This is because, function f_n maps 2^a elements to X number of elements ($X <= 2^a$) in U and function g_n maps 2^b elements to Y elements ($Y <= 2^b$) in U. This makes $X + Y <= 2^a + 2^b$.Since $X + Y = 2^n$ by definition of complement, $2^a + 2^b >= 2^n$.

3.3 Algorithm for complement construction

Input to a complement construction algorithm is the function f_n and the output is a complement function g_n of f_n.(Univariate function (polynomial) has been assumed. This can be generalized to multivariate polynomials.). Algorithm comprises of 3 high-level steps.

3.3.1 STEP 1-Getting the range set for complement function

From the function f_n we can get the set T = $\{f_n(i)/0 \leq i \leq 2^a - 1\}$.From this the set S = $U \quad \backslash \quad T$ can be constructed.

3.3.2 STEP 2-Construction of mapping

Before we construct a function representation, a mapping has to be established between a set $P = \{0, 1, 2, 3, ..., 2^b - 1\}$ and the set S. This mapping is a lookup table which imposes an ordering on the set S. We have that $|S| = 2^n - |T|$. For a valid function to exist, we must have $2^b \geq 2^n - |T|$ (since for a valid function, domain must be greater than or equal in size to range) and hence we have to choose b such that $b \geq log_2(2^n - |T|)$. Number of possible mappings(and hence functions) between P and S is $(|S|)^{|P|}$. But this includes mappings which are not onto also. To cover set S, we have to choose one onto mapping to extract a complement function out of various possible complement functions. One such mapping is done as follows

1. i=0
2. while($i <= 2^b$)
    - (a) If set S is not empty, choose uniformly at random, an element e from set S and map it to i and add to the lookup table as $< Key : Value >$ pair $< i : e >$. Remove element e from the set S. This implies $g_n(i) = e$ where g_n is a complement to be constructed.
    - (b) else if the set S is empty, map the first element in the ordering $g_n(0)$ to i by adding $< i : g_n(0 >$ to lookup table i.e $g_n(i) = g_n(0)$.
    - (c) i := i+1

The above mapping procedure selects one mapping out of $|S|!$ onto mappings (since after adding each element to lookup table, number of possible candidates reduces by 1 element from previous iteration). Many other ways of mapping are possible different from the procedure above. Each one of these mapping procedures result in different complement function.

3.3.3 STEP 3-Construction of complement function representation from mapping

Once we have the mapping constructed as in previous step, we can either apply 1) polynomial interpolation (or) 2) Arithmetization using simple replacement of boolean expressions with arithmetic expressions (or) by fourier polynomials 3) or by applying lambda calculus to get the function representation for a complement.These 3 algorithms for construction of complement function from mapping are explained in detail next.

Algorithm 1 - By polynomial interpolation

1. Since we have been given with a mapping from previous step between $x_i \epsilon \{0, 1\}^b$, and $y_i \epsilon S$, we can apply polynomial interpolation theorem which states that given $x_1, y_1, x_2, y_2, ..., x_n, y_n$ with $x_i \neq x_k$ for $i \neq k$, $\exists$ polynomial $p^{(n)} \epsilon F[x]$ of degree less than or equal to n-1, such that $p^{(n)}(x_i) = y_i$. The recurrence for building the polynomial is as follows:
$p^{(i+1)}(x) = p^{(i)}(x) + \{[y_{j+1} - p^{(i)}(x_{j+1})]/q^{(i+1)}(x_{j+1})\} * [q^{(i+1)}(x)]$
where $q^{(i+1)}(x) = \prod_{j=1}^{i}(x - x_j)$

Algorithm 2 - By representing boolean function by polynomial:

1. From the mapping obtained in previous section, we can construct function representation for a complement. Let binary of $x = s_b s_{b-1} ... s_3 s_2 s_1$ and binary of $Mapping(g_n)(x) = z_n z_{n-1} ... z_3 z_2 z_1$ obtained by looking up x from mapping lookup table constructed earlier. The i^{th} bit of $Mapping(g_n)(x)$ is denoted as $g_n^i(x)$.

2. Construct a DNF boolean function for z_i with 2^b clauses where each clause corresponds to a binary string in $\{0,1\}^b$ with binary digits replaced by variables as: $z_i = (\neg s_b \cap \neg s_{b-1} \cap ... \cap \neg s_3 \cap \neg s_2 \cap \neg s_1 \cap g_n^i(x_0)) \cup (\neg s_b \cap \neg s_{b-1} \cap ... \cap \neg s_3 \cap s_2 \cap \neg s_1 \cap g_n^i(x_1)) \cup ...(s_b \cap s_{b-1} \cap ... \cap s_3 \cap s_2 \cap s_1 \cap g_n^i(x_{2^b-1}))$. For example clause for a binary string 001010 with r=6 will be $(\neg s_6 \cap \neg s_5 \cap s_4 \cap \neg s_3 \cap s_2 \cap \neg s_1)$.The bits of $Mapping(g_n)(x)$ will be hard-coded in the formula above from mapping already constructed.

3. The boolean function above outputs the $i^{th} \quad bit \quad of \quad g_n(x) \quad where \quad binary(x) = s_b s_{b-1} ... s_3 s_2 s_1$. Intuitively, this boolean function is a multiplexer which selects the i-th bit for $g_n(x)$ among 2^b candidate bits. We have to construct n boolean functions as above for each of the n bits of $g_n(x)$

4. These boolean functions can be minimized to obtain minimum equivalent expression.After minimization we can arithmetize the boolean functions in one of the two ways below to get arithmetic expressions(polynomials):

   (a) Simple Arithmetization

      i. In the above boolean expression, replace a positive literal s_i by a variable s_i and a negative literal $(\neg s_i)$ by an arithmetic expression $(1 - s_i)$.

      ii. In the above boolean expression, replace a subexpression of the form $y_m \cap y_n$ by the arithmetic expression $t_m * t_n$

      iii. In the above boolean expression, replace a subexpression of the form $y_m \cup y_n$ by the arithmetic expression $1 - ((1 - t_m) * (1 - t_n))$

      iv. Repeat above steps recursively till the full boolean function for z_i gets arithmetized.

      v. At the end of above algorithm the whole boolean function for z_i is arithmetized as a polynomial in variables $s_b, s_{b-1}, ..., s_3, s_2, s_1$ which are the bit positions of input x. Let us denote this by $Arith(z_i)$.Since we have arithmetic expressions for individual bit positions of $g_n(x)$ in terms of bit positions of x, we can construct $g_n(x)$ by weighted summation given as $g_n(x) = 2^n * Arith(z_n) + 2^{n-1} * Arith(z_{n-1}) + ... + 1 * Arith(z_1)$. This is the required complement function $g_n(s_b, s_{b-1}, ..., s_3, s_2, s_1)$

   (b) Arithmetization by fourier expansion

      i. Since we have a boolean function for each bit position z_i in step 2, the RHS of each z_i can be expanded as a fourier polynomial in variables $s_b, s_{b-1}, ..., s_3, s_2, s_1$ which are the bit positions of input x.Thus we have a polynomial for i-th bit of $g_n(x)$ in terms of bit positions of x. Let us denote this expansion as $fourier(z_i)$

      ii. Thus $g_n(x)$ can be represented as the weighted summation of the bit positions of $g_n(x)$ as : $g_n(x) = 2^n * ((fourier(z_n) + 1)/2) +$

$2^{n-1}*((fourier(z_{n-1})+1)/2)+...+1*((fourier(z_1)+1)/2)$ since fourier polynomials evaluate to $\{-1, 1\}$ and this transformation is needed to get the bits $\{0, 1\}$

iii. Complement function $g_n(s_b, s_{b-1}, ...s_3, s_2, s_1)$ thus has been constructed by applying fourier representation ($g_n(x)$ is a polynomial in the bit positions of x ($s_b, s_{b-1}, ...s_3, s_2, s_1$))

Algorithm 3 - applying Lambda Calculus:

1. Obtaining $g_n(x)$ given x, suffices for showing the existence complement. First we develop an iterative procedure for returning $g_n(x)$ if x is given as follows:

2. Procedure GetGofX(x):

```
(a) u:=0
(b) z:=-1
(c) choose b such that $b \geq log_2(2^n - |T|)$ where T = $\{f_n(i)/0 \leq i \leq 2^a - 1\}$.
(d) if ( $x > 2^b - 1$) printerror "error: out of range"
(e) while($u \leq 2^n - 1$ and $z \leq 2^b - 1$)
    i. if u is in T then u:=u+1
    ii. if u is not in T then
        A. z:=z+1
        B. if($u \leq 2^n - 1$)
        C. {
        D. if(z equals x) add the entry $< x : u >$ to mapping table to
           signify $g_n(x) = u$ and return u as $g_n(x)$
        E. else add the entry $< z : u >$ to mapping table to signify
           $g_n(z) = u$ and set u:=u+1
        F. }
        G. else break
(f) if($u \geq 2^n$) then add the entry $< x : g_n(0) >$ to mapping table
    to signify $g_n(x) = g_n(0)$ and return $g_n(0)$ as $g_n(x)$ by looking up
    mapping table for key 0
```

3. Correctness of the above procedure can be proved since we are guaranteed that $2^b \geq (2^n - |T|)$ by choice of b. The while loop breaks when all the 2^n elements fall either into $L(f_n)$ or to $L(g_n)$ and similar to previous section on construction of a mapping, we hardcode $g_n(x) = g_n(0)$

4. The inequality $2^a + 2^b \geq 2^n$ must hold as mentioned earlier in section on conditions for existence of complement (where a and b are input sizes to f and g respectively).

5. Now that we have an iterative procedure for $g_n(x)$, we can convert this into lambda expression by replacing loops with a fixed point combinator from lambda calculus. The equals() relation above is bitwise equivalence represented by literals for each bit. This lambda expression can again be converted to a standard logical formula(for example a boolean formula) in terms of bit positions.

4 Existence of a complement when domain and range are infinite

4.1 Definitions

1. Function f is defined over the domain of integers and range of integers - f:$Z - Z$
2. Function g is defined over the domain of integers and range of integers - g:$Z - Z$
3. $L(f)$ = language generated by the function $f = \{f(x_0), f(x_1), ...)\}$ - an infinite set
4. $L(g)$ = language generated by a complement function $g = \{g(x_0), g(x_1), ...)\}$ - an infinite set
5. Function g is defined as *a complement* of f if
   (a) g is total
   (b) $g(x) \neq f(x) \quad \forall x$
   (c) $L(f) \cup L(g) = \{1, 2, 3, 4, 5, ...\}$ and
   (d) $L(f) \cap L(g) = \phi$
6. The subscript n for the functions f and g has been removed compared to previous section.

4.2 Decidability of complementation

To find an algorithm which would give a complement function g, given input function f.There are two cases to consider:

1. $L(f)$ is not recursive but recursively enumerable.
2. $L(f)$ is recursive

4.3 Case 1 - $L(f)$ is not recursive but recursively enumerable

Q - $L(f)$ recognizer

```
1. for all x
   (a) if $y == f(x)$ then return "y is in $L(f)$"
2. return "y is not in $L(f)$"
```

$L(f)$ is recursively enumerable since TM Q, outputs yes if a string y is in $L(f)$ ($y = f(x)$ for some x) but loops when y is not in $L(f)$. But $L(f)$ is not recursive since TM Q does not halt on all inputs. From definitions above we have complement of $L(f) = L(g)$. [Question of if Q outputs yes/no for any y is equivalent to asking if Q halts on any input. Reduction is defined as :

$x \quad \epsilon \quad L(Q) \Longleftrightarrow R(x) \quad \epsilon \quad HaltingProblem.]$ $L(f)$ is recursive if and only if both $L(f)$ and its complement $L(g)$ are recursively enumerable as proved below.

Claim: $L(f)$ is recursive if and only if both $L(f)$ and its complement $L(g)$ are recursively enumerable.

1. ($\Rightarrow$) If $L(f)$ is recursive then $L(f)$ is recursively enumerable. If $L(f)$ is recursive then complement of $L(f) = L(g)$ is recursive and hence $L(g)$ is recursively enumerable.

2. ($\Leftarrow$) Run two TMs M(f) and M(g) for $L(f)$ and $L(g)$. Input y. If y is in $L(f)$ then M(f) returns yes and M(g) loops. If y is not in $L(f)$ then M(f) loops and M(g) returns yes. Run M(h) accepting y which simulates M(f) and M(g). If M(f) return yes then M(h) returns yes and if M(g) returns yes then M(h) returns no. Thus M(h) returns yes or no for input y without looping.

Since $L(f)$ is not recursive, $L(g)$ is not recursively enumerable. So in this case, $L(g)$ has no turing machine that recognizes it.

4.4 Case 2 - $L(f)$ is recursive

$L(f)$ being recursive implies that there exists a turing machine which given input y always halts with accept($y \quad \epsilon \quad L(f)$) or reject ($else$) states.

If $L(f)$ is recursive , $L(g)$ is also recursive by closure under complementation. This implies $L(g)$ has a turing machine deciding membership.

4.5 Decidability of construction of a complement function for infinite recursive set $L(g)$ - for Case 2 above

In previous section, $L(f_n)$ and $L(g_n)$ were finite languages. This restriction was helpful for defining finite DNF formulas for complement g_n (or polynomial interpolation which is also for finite set of points). But for infinite recursive set $L(g)$, algorithms described in previous sections like polynomial interpolation or arithmetization do not work. This is because, we (either) might have to construct infinite boolean formulas and valuate such an infinite formula (or) might have to interpolate over infinite set of points. This procedure might never terminate. This can be proved undecidable by reduction from halting problem. Thus infinite cardinality for $L(g)$ makes construction of complement for infinite recursive sets undecidable.

4.6 Complementation as a nontrivial property - Application of Rice's theorem

1. Let Turing machine P_f which accepts an integer x as input and writes $f(x)$ in output and goes to accept state. P_f rejects if x is not an integer. $L(P_f) = L(f_n)$.. (A function can be thought of as a lambda expression which is equivalent to turing machine)

2. Let Turing machine Q_g which accepts an integer x as input and writes $g(x)$ in output and goes to accept state. Q_g rejects if x is not an integer.

$L(Q_g) = L(g)$.

3. Let Turing machine X_i which accepts a pair of encodings of Turing machines (P_f, Q_g). $L(X_i) = \{(P_f^k, Q_g^k)\}$, $k = 1, 2, 3, ...$ X_i accepts only if input is an ordered pair of turing machine encoding of 2 functions over integers, and rejects otherwise.

4. Let Turing machine Y which accepts encodings of Turing machines X_i and simulates X_i.
$L(Y) = \{X_i\}$, $i = 1, 2, 3, ...$

We want to test a property P such that language decided by X_i - the set of ordered pairs of TM encodings - are complements of each other. This property P is nontrivial since not all ordered pairs are in P and if $L(X_m) = L(X_n)$ then both X_m and X_n are in P. Since by Rice's theorem it is undecidable if a language decided by a Turing machine has a nontrivial property, L(Y) is undecidable.

5 References

1. Lecture Notes on Algebra and Computation, MadhuSudhan, MIT (for Polynomial Interpolation)

2. Various literature on Lambda Calculus on internet

3. A question posted to Google Group sci.math $http : //groups.google.com/group/$-

   (a) $/sci.math/browse\_thread/thread/46ab0335ce9205d9/f3c67ee19926e02e$

4. Complexity-I and Complexity-II Lecture Notes(2009 through 2010) - IMSc

5. Various literature on Rice's Theorem on internet

6. Automata Theory and Formal Languages - Aho, Hopcroft and Ullman

7. Introduction to Theory of Computation - Michael Sipser

TAC 2010 Summarization Track - Update Summarization with Interview Algorithm

Ka.Shrinivaasan, Chennai Mathematical Institute (CMI) (shrinivas@cmi.ac.in)

Abstract

Existing models for ranking documents(mostly in world wide web) are prestige based. In this article, alternative graph-theoretic schemes to objectively judge the merit of a document independent of any external factors (like link graph) and without probabilistic inference are proposed and application of these to TAC 2010 Update summary component is presented.

1 TAC 2010 dataset preprocessing and algorithms used

TAC 2010 dataset was split into candidate and reference sets. 25 out of 92 folders in the datasets were evaluated. In each folder, the datasets were split arbitrarily into reference and candidate texts. Both reference and candidate texts were concatenated to get two big documents - reference and candidate. These preprocessed texts were then applied to Interview algorithm described in detail below. Description of the algorithm is essential to understand how the dataset was evaluated to get intrinsic merit score and application of a threshold to this score to create summary. No guided summarization aspects were used in the TAC 2010 runs and focus was on update summarization component alone.

2 Motivation

Motivation for objective, independent judgement of a document is founded on the following example:

> Judge X decides about the merit of an entity Z purely by what other entities opine about Z without interacting with Z; Judge Y decides about the merit of Z by interacting only with Z. Question now is who is better judge - X or Y.

Probability of judgmental error of judge X is equal to probability of collective error of entities opining about Z while probability of judgemental error of judge Y is 0.5 as the following elementary arithmetic shows. Let us assume there are $2n$ voters and they need to decide/vote on whether a candidate is good or bad. A candidate getting majority ($n+1$ good votes) will be winner.

> Question: What is the probability that people have made a good decision?
>
> Answer: Probability of each voter making a good decision is p and bad decision is $1-p$ (0 <= p <= 1). Let p = 0.5 for an unbiased voter.

So for a candidate to be judged 'good', atleast $n+1$ people should have made a good decision. Probability of a good choice for these $2n$ voters, skipping the calculations, is :

$$\begin{aligned} P(good) = ((2n)!/4^n)* \\ ((1/((n+1)!(n-1)!)+ \\ 1/((n+2)!(n-2)!) \\ + ... \\ + 1/((n+n)!(n-n)!))) \end{aligned} \quad (1)$$

If there is an objective judgement without voting, probability of good decision is 0.5. It is interesting to see that above series tends to 0.5 as n grows infinitely. Thus, the judgement-through-majority-vote error probability is equal to the error probability of judge X who uses only the inputs from witnesses to judge Z while judgement-through-interaction(without election) error probability is equal to the error probability of judge Y (i.e. 0.5) who does not use witnesses. Thus, both judges X and Y are equally fallible but the cost incurred in a real world scenario for simulating X far outweighs that of Y.Thus it is worth delving into schemes for objective judgement like Y.

3 Three algorithms presented hereunder

1. Maxflow and Path lengths of Citation graphs - objective judgement (differs from Pagerank since it is Maxflow based and not prestige based)

2. Generalized Recursive Gloss Overlap - objective judgement (simulates judge Y with a 'white-box', invasive, intrinsic merit scoring) - covers majority of this report

3. Interview algorithm - objective judgement (simulates judge Y; Uses questions and answers to judge a candidate - 'black-box' and less-invasive - and also incorporates intrinsic merit score obtained from either MaxFlow of Citation graph or Generalized Recursive Gloss Overlap)

4 Directed Graph of Citations

4.1 Average Maxflow and Path lengths of Directed Graph of Citations

Given a corpus, algorithm constructs directed graph of incoming links to a document x from those documents chronologically later than x.Thus corpus is partitioned into set of digraphs. Indegree of a vertex in this digraph reflects the importance of a document represented by a vertex. This digraph can be thought of as a flow network where concept flows from a document to others which cite. Each edge has a weight. Capacity/weight for an (u,v) edge is defined as number of references v makes while citing u though there could be other ways to weight an edge. Assigning polarity to this capacity/weight is discussed in 4.2. Mincut of the digraph is the set of documents which are "potentially most influenced by the source document" (because maximum flow of concept from source occurs through this set to outside world/sink). Thus size of maxflow/mincut, averaged over all vertex-pairwise maxflow values, is a measure of influence of a source document in a community and thus points to its merit. (E.g., Chronology for web documents can be found by 'Last-modified' HTTP header which every dynamic document server is mandated to send to client). Alternative way to get the merit is to count the number of vertices in a predefined radius from source (i.e set of paths of some fixed length from source) which can be less accurate and sometimes misleading. Thus documents can be ranked using average Maxflow values. Advantage of this scheme is that it quantifies the extent of percolation of a concept within a community through Maxflow, without giving importance to the prestige measure of the vertices(documents) involved. So, this is one way of objectively assessing the merit of a vertex(document). Implementation applies Ford-Fulkerson algorithm to each s, t distinct pair and finds the average maxflow out of each vertex.

4.2 Polarity of citation edge

Parse the document/sentence containing the citation/link into tokens and find polarity. Whether a word is positive or negative can be decided by:

1. looking up a sentiment annotated ontology (e.g positivity/negativity of a lemma in SentiWordNet) or

2. entropy analysis - using $\sum_{i=0}^{1}(-P(i)logP(i))$ where P(0) = percentage of positive words and P(1) = percentage of negative words. Closer the entropy to zero, clearer the sentence/document on its viewpoint (very good or very bad) or

3. recursive gloss overlap algorithm to the citing document to get the polarity/sentiment of context citing the document.

Implementation tries all the three above. If the polarity/sentiment is negative, the weight for edge (u,v) is made negative in citation digraph, indicating a negative flow of concept to vertex v from the cited vertex u.

5 Definition Graph Convergence(or)Generalized recursive gloss overlap

5.1 Motivation for computing Intrinsic Merit of a document

Intrinsic merit is defined as the amount of intellectual effort put forth by the reader of a document and we try to quantize this effort. It is important to note that this quantized effort is independent of any observer/link-graph. Any document goes through some human understanding and we try to model it through what can be called Iceberg/Convergence/Generalized recursive gloss overlap algorithm (named so because a web document contains only a tip of the knowledge a document represents and understanding the document requires deeper recursive understanding of the facts or definitions the document is home to.).For example, going through a research paper requires the understanding of the concepts which draw a logical graph in our mind. Thus time spent on grasping the concepts and hence the intrinsic merit is proportional to the size and complexity of this graph and points to its merit (which is equal to the intellectual effort of the human reader). Since WordNet is the existing model for semantic relationship, we will try to establish that a text document can be mapped to a graph which is a subgraph of WordNet and merit can be derived applying some metrics on this graph. This is the intuition behind the algorithms that follow.

5.2 Definition tree of a document

Given a document its definition tree is recursively defined as

Definition 1. *definitiontree(all keywords of document) = definitiontree(term1) definitiontree(term2) ...definitiontree(termn) where term1, term2,...termn occur in the definition of keywords of a document.*

For example, let us consider the following document which talks about Kuratowski theorem

> Document1 = Every K5,K3,3-free graph is planar

This document contains key terms like "K5,K3,3-free", "graph" and "planar".Now we recursively construct the definition tree for these terms. Key terms are decided after filtering out stopwords and by computing TF-IDF and only terms above a threshold tfidf are chosen for constructing the definition graph.

definitions at level 1:

1. K5 = Complete graph of 5 vertices (key terms: graph, vertices)

2. K3,3 = graph of two sets of 3 vertices each interconnected (key terms: graph, two sets, vertices, interconnected)

3. graph = set of vertices and edges among them (key words: vertices, edges, set)

4. planar= graph embedded on a plane (key words: graph, embedded, plane)

Thus the definition tree goes deeper as each keyword/concept is dissected and understood. Given above is level-1 grasping of the document. Important thing to note is that intersection of the sets of keywords in the definition of K5, K3,3, graph and planar is not an empty set(glosses for two or more keywords overlap). For example, intersection of definitions of K5 and K3,3 is the set {graph, vertices}. Thus the overlap of the terms "graph" and "vertices" in two definitions of K5 and K3,3 is an indication of deeper cohesion/interrelatedness of the terms in the document.Thus the replicated terms(represented by vertices) in the definition tree can be merged to get convergence (gloss overlap generalized to more than two glosses). Thus the definition tree is transformed into definition graph(since a vertex can have more than one parent) by merging replicated keyword vertices into 1 vertex. Synset definitions in WordNet gloss are used for getting keyword definitions in the implementation. But WordNet Gloss does not work for terms specialized for a domain(e.g gloss for "graph" does not have a synset for graph theory as part of its senses set). This requires ontologies for the class the document belongs to. Thus recursive gloss overlap algorithm is limited by WordNet in present implementation. At each level, word sense disambiguation is done by following Lesk's algorithm adapted to Generalized Recursive Gloss overlap to choose the synset definition fitting the context. It is important to note that 1) only one relation ("is in definition of") is used and 2) only keywords within the document are considered 3) gloss overlap is computed recursively at each level of understanding till required depth is reached.

5.3 Definition graph convergence and steps of Recursive Gloss Overlap algorithm

Convergence of a document is defined as the decrease in the number of unique vertices of the set of definition trees of its keywords from level k to level k+1 For example definition tree of the above document converges to {edges, vertices} after expanding the definition tree further down. Thus the above document has "edges" and "vertices" as its undercurrent. Thus the Convergence algorithm takes no labelled examples for inference. Only requirement is to have a dictionary/gloss/ontology of terms and their corresponding definitions. If a documents definition tree does not converge within a threshold called "depth" number of levels then the document is most likely less meaningful or of low merit. Thus the Convergence algorithms strikingly adapts an iceberg which has seemingly unconnected set of "tips" at the top but as we go deeper get unified. Level where this unification happens is a differentiator of merit. If while recursively expanding the definition tree, a vertex results in a child vertex which is same as some sibling of the parent then we compute and remove the intersection of keywords at present and previous level - since these common vertices have already been grasped. Accordingly, number of edges, vertices and relatedness are updated for each level. Number of vertices are adjusted for removal of common tokens, but number of edges remain same since they just point to a different vertex at that level. This process continues top-down till required depth is reached.

Steps:

1. Get the document as input
2. currentlevel = 1
3. keywordsatthislevel = {keywords from the document through tfidf filter (e.g > 0.02)}
4. While (currentlevel < depthrequired) {
    - For each keyword from keywordsatthislevel lookup the best matching definition for the keyword and add to a set of tokens in next level - requires WordSenseDisambiguation - implementation uses Lesk's algorithm
    - Remove common tokens with previous levels since they have been grasped in previous level (this is an optimization)
    - Update the number of vertices, edges and relatedness (vertices correspond to unique tokens, edges correspond to the single relation 'y is in definition of x' and relatedness is linear overlap or quadratic overlap) and Update tokensofthislevel
    - currentlevel = currentlevel + 1

    }
5. Output the Intrinsic merit score =

    $$|vertices| * |edges| * |relatedness| / firstconvergencelevel \quad (2)$$

    Where

    - Relatedness =$NumberOfOverlaps$ (linear, also called as convergence factor) (or)
    - Relatedness =

      $$NumberOfOverlappingParents * NumberOfOverlaps^2 \quad (quadratic) \quad (3)$$

    - firstconvergencelevel = level of first gloss overlap

At the end of recursive gloss overlap, nodes with high number of indegrees(parents) are indicators of the class of the document since greater the indegree, greater is the number of keywords overlapping (voting for an underlying theme).From graph theoretic view, Definition Graph constructed above is a multipartite graph since vertices can be partitioned into sets with no edges within a set and edges only across sets(without removal of common tokens between levels - which is only an optimization since by removing common tokens we redirect edges to vertices within the same set and multipartiteness is lost). Preserving multipartiteness is useful since it groups the tokens at each level of recursion into single set with edges across these sets - multipartite cliques of this multipartite graph can be analyzed to get the robustness. Moreover, this algorithm ignores grammatical structure. Reason is that principal differentiator in analyzing relative merit of two documents is the quality of content and complexity of content and both documents are equally grammatical.Quality of content is proportional to the vertices of the definition graph and complexity of the content is proportional to the relatedness and edges of definition graph. In spite of ignoring grammatical structure, the graph constructed above is context-sensitive since word sense disambiguation is done while choosing the synset matching a keyword. This way, the definition graph is a graph representation of the knowledge in the document sans the grammatical connectives.

5.4 Definition of shrink

Definition 2. *Let us define "shrink" to be the amount of decrease in the number of unique vertices between levels k and $k+1$ during convergence (gloss overlap)*

5.5 Comparison of two documents for relative merit - two examples

Document1 : Car plies on sky

Constructing definition graph for level-1 we get,

1. Car - automobile used for surface transport
2. plies - is flexible; goes on a surface; moves
3. sky - atmosphere; not on earth;

As can be readily seen there is overlap of 2 key terms at level 1 of the tree and thus there is less gloss overlap. Thus at level-1 document looks less meaningful.

Document2 : Cars and buses ply on road

Constructing definition graph for level-1 we get,

1. Car - automobile used for surface transport
2. Buses - automobile used for surface transport
3. ply - flexible; go on a surface; move
4. road - asphalted surface used for transport

All 4 keywords overlap giving surface as common token in their respective glosses. Overlap is better than Document1, since more keywords contribute to overlap.Both examples are grammatically correct but one of them is less related semantically.

5.6 Intrinsic merit score, Convergence factor and Relatedness

Definition 3. *Let us define Intrinsic merit I to be the product of number of vertices(V), number of edges(E) and Convergence factor(C) of the definition graph of the document.*

$$I = V * E * C \tag{4}$$

Convergence factor (C) is the difference between number of vertices in definition tree and number of vertices in definition graph (V). Number of vertices in definition tree includes overlapping vertices without coalescing them (since after coalescence we get the definition graph).Number of vertices in the definition tree $= x^d - 1$ where x is the average number of keywords per term definition and d is the depth of the definition tree of the document. Let us add 1 to this to get x^d (smoothing). Number of vertices in the definition graph = V Thus the Convergence factor C and Intrinsic merit I become,

$$C = x^d - V \tag{5}$$

$$I = V * E * (x^d - V) \tag{6}$$

Intrinsic Merit score can also be further fine-tuned by taking into account the level of definition tree at which first convergence(gloss overlap) happens, defined as firstconvergencelevel. Greater the firstconvergencelevel, more irrelevant the document "looks" (but has a deeper cohesion). Depth to which definition tree has to be grown is decided by extent of grasp needed by the reader. Thus greater the depth of definition tree, greater is the understanding. It is obvious to see that Depth has to be greater than firstconvergencelevel so that some pattern can be mined from the document. Heuristically, we can grow the definition tree till intersection of leaves of all sub-trees of the keywords in the document is non-empty. This is the point where we can safely assume that all keywords in the document have been somehow related to one another. So, Intrinsic merit score can be improved by incorporating firstconvergencelevel denoted by f. Thus improved score is

$$I = V \cdot E \cdot (x^d - V)/f \tag{7}$$

(since merit is inversely proportional to firstconvergencelevel) .Complexity of constructing definition tree is O(x^d). Since non-unique vertices are coalesced(through gloss overlap), definition graph can be constructed in O(V) time (subexponential). Since x is the average number of children keywords per keyword, $x = E/V$. Substituting,

$$I = E * V * (E^d - V^d)/(V^d * f) \tag{8}$$

As an alternative to convergence factor, gloss relatedness score similar to the one discussed by Banerjee-Ted, but considering only one relation, number of overlapping parents and length of overlap can be used to get the interrelatedness/cohesion of the document. Replacing the convergence factor with relatedness, Intrinsic merit becomes, $I = V \cdot E \cdot \mathit{Rel}/f$ where Rel is the sum of relatedness scores, computed over all overlapping glosses at each convergence level and f is the level at which first gloss overlap occurs

$$\begin{aligned} \mathit{Rel} = \sum_{i=1}^{n} (&\mathit{relatedness}(\mathit{Level}(i), \\ &\mathit{keyword}1, \mathit{keyword}2, ..., \mathit{keywordn})) \end{aligned} \tag{9}$$

This relatedness score has been generalized to overlap of more than two glosses with single relation R (R(x,y) = y is in definition of x). Function relatedness() for n-overlapping keywords is defined as,

$$\begin{aligned} &\mathit{relatedness}(\mathit{Level}(i), \\ &\quad \mathit{keyword}1, \mathit{keyword}2, ... \\ &\quad , \mathit{keywordn}) = \\ &\quad \mathit{OverlapLengthAtLevel}(i) \\ &\quad (\mathit{LinearOverlap}) \end{aligned} \tag{10}$$

(or)

$$relatedness(Level(i), keyword1, keyword2, ... , keywordn) = n \cdot (OverlapLengthAtLevel(i)^2) \quad (QuadraticOverlap) \tag{11}$$

The relatedness score reflects the convergence since it takes into account the overlapping keywords at each level and length of the overlap. Thus first version of relatedness() function, implies the convergence factor (difference in number of vertices of definition tree and definition tree, signifying overlap) Intrinsic merit/Relatedness score can be used to rank the set of documents and display them to the user. Referring back to examples in 5.5, quadratic relatedness measure ((9) above) is a better choice than linear overlap since it is a function of both overlapping parents and the overlap length. The quadratic overlap gives greater weightage to length of overlap by squaring it while keeping the number of parents involved linear.

5.7 Intuition captured by above intrinsic merit score

The number of edges (representing relation between parent term and its definitions) increase as relationship among vertices of definition graph increases. The number of vertices(keywords) in the definition graph increases, as the knowledge represented by the document increases. The depth of the definition tree increases, as the understanding grows. Convergence factor increases as number of overlapping terms in definition graph increases. Similarly quadratic relatedness score increases with number of keywords involved in overlap and the length of overlap, thus pointing to stronger semantic relationship among the keywords. Intuitively, definition graph is WordNet(or any other ontology) projected onto the document.

5.8 Breadth/Depth first search of definition graph and why it is not a good choice for computing merit score

Since Breadth/Depth first search of graph can model human process of thinking, BFS/DFS algorithms can be applied to get the merit score. Since BFS/DFS algorithms run in O($V + E$) time merit score is proportional to $V + E$ - all vertices of the graph are visited in O($V + E$) time. But the drawback of this approach is that strength of underlying theme of the document and cohesion of keywords is not captured by this merit score. Since Intrinsic merit score obtained by Convergence reckons with depth and overlapping keywords, BFS/DFS merit score is discarded

5.9 Sentiment analysis applying Recursive gloss overlap

Recursive Gloss Overlap algorithm after few levels down the definition tree would spell out the sentiment of writer.

> Example1: "That movie was fantastic; Graphics was awesome" Keywords at level-1 of Definition graph construction:
>
> 1. movie - motion picture; positive
> 2. fantastic - good, excellent; positive
> 3. graphics - software technique; positive
> 4. awesome - good, great; positive

Overlapping terms are {good, positive} and large number of keywords(parents) contribute to this overlap. Thus the document is of extolling nature about some target entity. Prerequisite is a dictionary which annotates each word with the sentiment and sense of the word(Implementation uses SentiWordNet which gives positivity/negativity for each lemma). Sentiment analysis with Recursive Gloss Overlap is applied to finding the polarity of an edge in Citation graph (See (1)). Recursive Gloss Overlap algorithm is applied to each Citation context and a definition graph is constructed. Keyword vertices with more than one indegree are then tested for positivity and negativity using SentiWordNet. If majority of these is positive then polarity for citation edge is positive, otherwise negative.

5.10 False negatives

Convergence algorithm never assigns lower merit score to a document which deserves a higher merit since a document with higher merit explains the concept with more depth/cohesion than document with lower merit. So false negatives do not exist

5.11 False positives

False positives exist since both a document and its arbitrarily jumbled version will get same merit score. This is prevented by assuming grammatically correct documents or by preprocessor which does parts of speech parsing to validate the grammatical structure of the document.

5.12 Definition graph and Hyperlink graph

Prestige measures obtained from hyperlink graph for a given document are dependent on prestiges of linking documents whereas the Definition graphs are results of human judgements in different viewpoint (e.g WordNet is a result of some experiments done on human judgements). Moreover the hyperlink graph is coarse-grained interconnection of documents and the Definition graph is fine-grained interconnection of words within the same document. Definition graphs are projections of a larger, absolute, universal graph (e.g WordNet). Thus definition graphs depend only on the accuracy of this absolute ontology of which they are subgraphs and definition graphs place one more level of abstraction on the way "judgement" is perceived. We can imagine this to be a two phase process - 1) electing a system which in turn judges documents objectively (e.g WordNet is the elected system) 2) judgement of a document by the elected system (e.g application of WordNet to judge a document as in definition graph construction).

5.13 Normalization

Intrinsic merit can be compared only if the compared documents are of same class. Thus 2 documents explaining special relativity can be compared while a document on journalism can not be compared with a document on special relativity. Intrinsic Merit scores can be normalized by,

$$NormalizedIntrinsicMeritScore = Score/MaximumScore \quad (12)$$

5.14 Ordering and Relative Merit

Definition 4. *Document1 is more meritorious than document2 if*

1. *document1 has more keywords that need to be understood than those of document2,*

2. *cohesion/interrelation of the keywords in document1 is more than that of document2,*

3. *average number of keywords per definition is greater for document1 than document2,*

4. *firstconvergencelevel(level at which first gloss overlap occurs) of document1 is less than that of document2 and*

5. *depth of definition tree of document1 is greater than that of document2.*

If we want a weaker definition of the above, ranking may be a partial order(where some pairs of documents may not be comparable) than a total order.This appeals to intuition since document1 may be better in some aspects but worse in some other relative to document2

5.15 Semantic relatedness or Meaningfulness of a document

Definition 5. *A document is meaningful to a human reader if any pair of keywords in the document are within a threshold WordNet distance e.g Jiang-Conrath distance*

5.16 Formal proof of correctness of Convergence and Intrinsic Merit Score

Theorem 1. *If a document lacks merit, convergence(or gloss overlap) does not occur (Corollary: Document's merit is measured by extent of convergence)*

Proof. By "meritorious" document, we imply a document which is meaningful as per the definition of meaningfulness above(i.e. keywords in a document are separated within threshold WordNet distance metric like Jiang-Conrath distance). Let us denote R as a relation "is descendant of". If xRy then y is in (gloss)definition tree of keyword x(i.e y is descendant of x). If definition trees of keywords of the document are disjoint, then there is no y such that xRy and zRy for two keywords x and z. Let us define the relation S to be "two keywords are related". xSz iff xRy and zRy for some y. Thus we formalise cohesiveness/meaningfulness of a document in terms of definition graph. If a document is not meaningful then there exist no x and z such that xSz, which implies that for no y, xRy and zRy. Thus there exist no vertex y which is in definition tree of two key words. Thus convergence is a necessary condition for merit. The relation S implies that there exists a path between two keywords x and y in the document, through some intermediate nodes which are in the definition/gloss tree of x and y. There exists a threshold WordNet distance greater than length of this path since the length is finite and whether a document is meaningful depends on this threshold. Thus convergence(generalized gloss overlap) implies meaningfulness of a document as per the definition above. Moreover Intrinsic merit increases with number of edges and relatedness() - linear or quadratic. So with greater relatedness() and more number of vertices and edges, overlaps and number of nodes involved in overlap increase. This in turn implies that more number of paths are available amongst the keywords of the document since every overlap acts as a meeting point of two keyword definition trees. Probability that lengths of these paths are less than threshold WordNet distance is inversely proportional to firstconvergencelevel(level of first gloss overlap) as follows. Probability that a path exists from x to y in the definition graph(P1) =

$$NoOf(Overlaps(DefTree(x), DefTree(y)))/ TotalNoOf(paths). \quad (13)$$

Probability that such an x-y path is less than the threshold WordNet distance (P2) =

$$NoOf(x-y \quad paths < \quad ThresholdLength)/ NoOf(x-y \quad paths) \quad (14)$$

Probability $P3 = P1 * P2$ (by conditional probability that there is a path between x-y and such a path is less than threshold length) is proportional to meaningfulness by definition above. With greater the first level in which gloss overlap occurs, the length of x-y path increases for all of the x-y paths penalising meaningfulness, since any x-y path has to pass through such an overlapping vertex due to multipartiteness.Thus intrinsic merit score discussed earlier captures this notion. □

5.17 Extending the above theorem for general graphs

Above theorem can be extended to general graphs by constraining the longest shortest path (diameter) of any pair of vertices (s,t) of the definition graph to be less than the threshold wordnet distance.But ranking scheme has to be re-invented since above ranking is specific to multipartite definition graphs.

5.18 Worst case running time analysis of Recursive Gloss Overlap algorithm

Let overlap at level i = OL(i) and branching degree = x (=average number of tokens per keyword gloss) Number of vertices in definition graph

$$V = x + x^2 + ... + x^z - \sum_{i=1}^{z} OL(i) \quad (where \quad z = (d-1)) \quad (15)$$

Running time for:

1. finding overlaps at level i and merge them to single vertex =

$$O(x^k)(where \quad k = 2 * i + 1) \quad (16)$$

2. get tokens =

$$O(x^i - OL(i)) \quad (17)$$

3. remove isomorphic nodes across levels =

$$O(x^k)(where \quad k = 2 * i + 1) \quad (18)$$

Steps 1) ,2) and 3) together have running time $O(x^p)$ where $p = 2d + 1$. But $V = O(x^d)$. Thus running time of recursive gloss overlap = $O(E * V^2)$ since x is upperbounded by V, where V is the number of nodes in Definition Graph and E is the number of edges in Definition graph.

5.19 Parallelizability

Recursive gloss overlap is parallelizable by partitioning the tokens at each level and assigning each subset to different processors (Map) to get the tokens for next level. Individual results from processors are merged (Reduce) to get the final set of tokens for a level. This is repeated for all levels. MapReduce can be applied for parallelism.

6 Interview Algorithm (applying (1) and/or (2) for computing intrinsic merit)

6.1 Motivation for Interview algorithm

Here we map the real world scenario of an interview being conducted on a candidate where a panel asks questions and judges the candidate based on the quality of answers by candidate - candidate is a document and it is "interviewed" by a reference set of authorities. Each document x is interviewed/evaluated by set of reference documents which will decide on the merit of the document x. Reference set initially consists of n user chosen authorities on the subject. Interview is set of queries made by reference set on the document and evaluating the answer to the queries. If x passes the interview it is inducted into reference set. Next document will be interviewed by n+1 documents including last selected document and so on. Hierarchy of interviews can be built. For example Document x interviews documents y and z. Document y interviews w and document z interviews p. Thus we get a tree of interviews (it could be a directed acyclic graph too, if a candidate is interviewed by more than one reference, one of which itself was a candidate earlier). The interview scores can be weighted and summed bottom-up to get the merit of the root (Analogy: hierarchy in an organization).

6.2 Steps of the Interview algorithm

1. Relevance of the document to the reference set is measured by a classifier (NaiveBayesian or SVM or search engine results for a query)
2. Intrinsic merit score of the document is computed either by Recursive Gloss Overlap algorithm (measures the meaningfulness/sanity of the candidate) (or) by citation digraph
3. Reference set interviews the candidate and gets the score
4. Value addition of the candidate document is measured (what extra value candidate brings over and above reference set)
5. Candidate is inducted into reference set based on the above criteria if candidate is above a threshold.

6.3 Mathematical formulation of an interview

Interview is abstracted in terms of a set of tuples, where each tuple is of the form

$$t(i) = (question, answer, expectedanswer, score) \tag{19}$$

for question i.

$$Interview(I) = \{t(1), t(2), t(3), ..., t(n)\} \tag{20}$$

$$t(i).score = PercentageOfMatch(t(i).answer, t(i).expectedanswer) \tag{21}$$

$$if(\sum_{i=1}^{n}(t(i).score)) > referencethreshold \tag{22}$$

then induct the document into reference set.In the Information Retrieval context, a question is a query and the answer is the context within the document that matches the query. The answer returned by the document is then compared with expectedanswer. Comparison is done by Jaccard coefficient of shingles (n-grams)

$$t(i).score = |shingle(answer) \bigcap shingle(expectedanswer)| / |shingle(answer) \bigcup shingle(expectedanswer)| \tag{23}$$

1. Supervised: In supervised setting, each reference document is pre-equipped with user-decided set of queries and answers it expects. Thing to note is that a document is made a live object - it both has content and questions it intends to ask(set of search queries). Alternative way to compute t(i).score is to find out the definition graph of answer and expectedanswer and compute the difference between the two graphs(e.g edit distance). Downside of this is the assumption of pre-existence of correct answers which makes this a supervised learning.
2. Unsupervised: In the absence of reference questions and answers, questions that a document "intends" to ask can be thought of as set of queries for which the document has better answers(results). These set of queries/results (questions and answers) can be automatically obtained from a document through an unsupervised way by computing set of more likely to be important n-grams(by computing key phrases with tfidf above threshold) and the context of the n-grams in the reference documents.These n-grams/contexts can later be used as "references questions" (n-grams) and "reference answers"(contexts of the corresponding n-grams) to the candidate document. Thus we compensate for lack of reference Questions and Answers. Alternatively, an interview can be simply considered as the percentage similarity of definition-graph(reference) and definition-graph(candidate) obtained by edit distance.

6.4 Searching for answer to a query within the document (as implemented)

If a document describing tourist places is given and the query is "What are the good places to visit in this city?", then query is parsed into key words like "good", "places", "visit" and "city" and matching contexts within the document are returned where context is the phrase of lenth $2n + 1$ (from $x - n$ to $x + n$ locations with location of keyword being x).

6.5 Value addition measure

Recursive gloss overlap algorithm gives the definition graph of the candidate document. To measure the value addition we can run the recursive gloss overlap algorithm on the reference set to get the definition graph of reference set and find out the difference between the two definition graphs - reference and candidate. Since value addition is defined as the value added which is not already present, extra vertices and edges present in candidate but absent in the reference set are a measure of value addition. Value addition can be measured by either edit distance(cost of transforming one graph to the other after adding/deleting vertices/edges), maximum common subgraph or difference of adjacency matrices. Implementation uses graph edit distance measure.

6.6 Update summarization through Interview algorithm (applying algorithm given in 6.2)

Given a news summary and a candidate news to be added to summary

1. Label the summary as reference set.

2. Run a classifier on summary and candidate to get the class to which both belong to (or get from search engine results on a news topic)

3. If class(summary) $==$ class(candidate) proceed further

4. Calculate intrinsic merit score of the candidate news document through recursive gloss overlap algorithm described in (2) (or) from citation digraph described in (1)

5. Candidate news is interviewed by reference set(summary in this case)

6. Compute value addition of candidate to summary

7. Add the value added information from candidate into existing summary to get new summary (by getting cream of sentences with top sentence scores)

6.7 Application to Topic Detection, Link detection and Tracking

Interview algorithm can be applied to TAC 2010 topic detection tasks though no runs were done specifically for this purpose.

1. Interview algorithm and graph edit distance measure can be applied to news topic link detection(Anwers the question - Does a pair of news stories discuss same topic?). Since same news item falls under multiple topics and is changing over time, topic of a news story is in a state of flux. Given a pair of news stories (n1, n2) execute interview of n2 with n1 as reference. This interview score decreases and edit distance grows as n2 becomes more irrelevant to n1. By defining a threshold for interview and edit distance scores to belong to same topic, link detection can be achieved. It is important to note that interview score and value addition score are inversely related.

2. At any point in time, compute edit distance for all possible pairs Nx, Ny in a topic (after getting their respective definition graphs) and choose Ny which has largest edit distance to others and hence an outlier and least likely to be in the topic. Thus topic detection is achieved(Answers the question - Does this story exist in correct topic?).

3. Topic tracking can be done by constructing definition graph and finding vertices with high number of indegrees. These keywords are voted high and point to the maximum likely topic of the news story (works as an unsupervised text classifier).This process has to be periodically done since topic of a story might change and thus the definition graph will change.

6.8 TAC 2010 Dataset Evaluation Methodology

1. Split each dataset into two : Reference and Candidate (as described in preprocessing section)

2. Compute the intrinsic merit score for Candidate:

- applying citation digraph construction (or) recursive gloss overlap - recursive gloss overlap was applied since it was difficult to get a citation graph for dataset
- Parse into keywords and get keywords above a threshold tf-idf
- Perform WSD using Lesk's algorithm
- Get glosses of matching sense through wordnet api
- get overlaps at level i, update intrinsic merit score (either using linear or quadratic overlap)
- repeat for sufficient number of levels defined by "depth"

3. execute interview if reference questions and answers are available (supervised) or through getting important n-grams/context from Reference by algorithm described above (unsupervised) - at present restricted to 1-gram for keywords and bigrams for jaccard coefficient calculation

4. compute value addition through definition graph edit distance between reference and candidate, and get the score.

5. get percentage weighted sum of intrinsic merit, interview and value addition scores and get final score.

6. APPLY (2), (3) and (4) ABOVE TO UPDATE SUMMARIZATION: If final score is above threshold, update the summary with candidate and publish top 5 percent of the sentences (sentence scoring is done by sum of tfidf scores of words in a sentence)

6.9 Results

25 out of 92 datasets were evaluated with interview algorithm described above. Some of the resultant summaries crossed 100-word limit but they were in the top 5 percent of the sentence scores. Results are as published in Guided Summarization Evaluations.

6.10 Conclusion

Results above demonstrate the application of interview algorithm to TAC 2010 update summarization task. Motivation for this excercise is to explore the possibility of finding a framework to assess the merit of a document with and without link graph structure in place with greater emphasis on the latter. Citation graph maxflow measures the penetration of a concept (represented in a document), in a link graph while the Recursive gloss overlap objectively judges the document without getting inputs from any incoming links. Interview algorithm uses either of these two algorithms and abstracts some real world applications. Moreover the intrinsic ranking scheme given above need not be the only possible way of computing merit. Once we have definition graph for a document (whether multipartite or not), multitude of more ranking schemes can be invented - for example 1) modelling the definition graphs as expander graphs 2) k-connectedness of the definition graph 3) (multipartite) cliques of (multipartite) definition graph etc., Since definition graph construction is computationally intensive, there is a scope of improvement in improving the recursive gloss overlap algorithm by applying some parallel processing framework like MapReduce. Applying Evocation WordNet, implementing a MapReduce(e.g Hadoop) cluster and considering more than one relation are future directions to think about. Theoretical foundation for the recursive gloss overlap comes from WordNet itself which visualises the relatedness of words - Definition graph is just an induced subgraph of WordNet for a document. Accuracy of Recursive gloss overlap depends on the accuracy of WordNet, depth to which definition trees are grown and Word Sense Disambiguation.

7 Acknowledgements

Algorithms discussed in this article were part of author's master's thesis done during December 2009 to July 2010. Author would thank Professors B.Ravindran (Indian Institute of Technology, Chennai, India) and Madhavan Mukund (Chennai Mathematical Institute, Chennai, India) for guiding through and encouraging me to participate in TAC 2010 and above all submit to God for granting intuition.

References

[1] Graph Similarity, Master's thesis by Laura Zager and George Verghese EECS MIT 2005

[2] Edit distance and its computation, presentation by Joszef Balogh and Ryan Martin

[3] Extended Gloss Overlaps as a measure of semantic relatedness, Satanjeev Banerjee and Ted Pedersen

[4] Sematic Language Models for TDT, Ramesh Nallapati,University of Amherst

[5] WordNet Evocation Project- http://wordnet.cs.princeton.edu/downloads /evocation/release-0.4/README.TXT

[6] SentiWordNet - http://sentiwordnet.isti.cnr.it/

[7] WordNet - http://wordnet.princeton.edu

[8] Navigli, R. 2009. Word sense disambiguation: A survey. ACM Comput. Surv. 41, 2, Article 10 (February 2009)

[9] Temporal information in Topic Detection and Tracking - Juha Makkonen, University of Helsinki

[10] Overview NIST Topic Detection and Tracking by G. Doddington , http://www.itl.nist.gov/iaui/894.01/tests/tdt/tdt99/ presentations/index.htm

[11] Topic Detection and Tracking Pilot Study - James Allan , Jaime Carbonell , George Doddington , Jonathan Yamron , and Yiming Yang UMass Amherst, CMU, DARPA, Dragon Systems, and CMU

[12] The cognitive revolution: a historical perspective , George A. Miller Department of Psychology, Princeton University, TRENDS in Cognitive Sciences Vol.7 No.3 March 2003

[13] On Bipartite and Multipartite Clique Problems, Milind Dawandeb, Pinar Keskinocakc, Jayashankar M. Swaminathan and Sridhar Tayur,Journal of Algorithms 41, 388âĂŞ403 (2001)

[14] Python Natural Language Toolkit - http://nltk.sourceforge.net

[15] Partitioning CiteSeer's Citation Graph - Revised Version , Gregory Mermoud, Marc A. Schaub, and Gregory Theoduloz, School of Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne (EPFL), 1015 Lausanne, Switzerland

[16] Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, Alexander Budanitsky and Graeme Hirst, Department of Computer Science, University of Toronto

[17] The Official Python Tutorial - http://docs.python.org/tut/tut.html

[18] MapReduce: Simplified Data Processing On Large Clusters, Jeffrey Dean and Sanjay Ghemawat, Google Inc.,

[19] Opinion Mining and Summarization - Sentiment Analysis , Bing Liu Department of Computer Science , University of Illinois at Chicago , Tutorial given at WWW-2008, April 21, 2008 in Beijing WWW 2008

[20] Web Data Mining, Bing Liu, Department of Computer Science , University of Illinois at Chicago

[21] Introduction to Algorithms - Second Edition, Thomas H.Cormen, Charles E.Liecerson, Ronald L.Rivest, Clifford Stein

Srinivasan Kannan (alias) Ka.Shrinivaasan (alias) Shrinivas Kannan

Krishna iResearch - nonprofit opensource initiative

Algorithms
Complexity
Theoretical Machine Learning
Logic
Number theory and everything else

TITLE	CITED BYYEAR
Indepth analysis of a variant of Majority Voting and relation to Zermelo-Fraenkel Set Thoery With Axiom of Choice (ZFC) K Shrinivaasan	2013
Decidability of Existence and Construction of a Complement of a given Function Shrinivaasan. Ka, Meena Mahajan arXiv preprint arXiv:1106.4102	2011
Few Algorithms for ascertaining merit of a document and their applications Shrinivaasan. Ka, Balaraman Ravindran, Madhavan Mukund arXiv preprint arXiv:1006.4458	2010
Update summarization with Interview Algorithm - http://www.nist.gov/tac/publications/2010/participant.papers/CMI_IIT.proceedings.pdf Shrinivaasan. Ka, Balaraman Ravindran, Madhavan Mukund NIST TAC 2010	2010

ka.shrinivaasan@gmail.com

home Updated Apr 6, 2018, 3:34 PM

Share

ஸ்ரீநிவாஸன் கண்ணன்

K.Srinivasan

(also known as: **Srinivasan Kannan, Shrinivaasan Kannan, Shrinivas Kannan)**

Shortest CV - LatexPDF

Longest CV - PDF

Longest CV - Text

Krishna iResearch Open Source Products - open source research initiative of self - (2003 - Present) - Profile (SourceForge), Profile (GitHub)

NeuronRain - a new Machine Learning, Cloud and Messaging augmented OS:

NeuronRain Research - SourceForge Repositories - academic research and astronomy oriented:

ACADPDRAFTS - Publications, Drafts etc.,

VIRGO - VIRtual Generic Os - Linux kernel fork-off (kernel modules,system calls etc.,) with cloud and machine learning features (32 bit based on 4.1.5 mainline)

VIRGO64 - 64 bit version of previous VIRGO Linux kernel based on 4.13.3 mainline

USBmd - USB driver kernel module for network analytics, debugging etc., based on 32-bit 4.1.5 linux kernel

USBmd64 - USB driver kernel module for network analytics, debugging etc., based on 64-bit 4.13.3 linux kernel

ASFER - AStroinFER - BigData Analytics and Machine Learning Software for Large Data Sets (at present implemented for Astronomical Datasets)

KINGCOBRA - Linux kernelspace messaging with miscellaneous applications based on 32-bit 4.1.5 linux kernel

KINGCOBRA64 - Linux kernelspace messaging with miscellaneous applications based on 64-bit 4.13.3 linux kernel

NeuronRain Enterprise - GitHub Repositories - for enterprise cloud deployments:

VIRGO - VIRtual Generic Os - Linux kernel fork-off (kernel modules,system calls etc.,) with cloud and machine learning features (32 bit based on 4.1.5 mainline)

VIRGO64 - 64 bit version of previous VIRGO Linux kernel based on 4.13.3 mainline

USBmd - USB driver kernel module for network analytics, debugging etc., based on 32-bit 4.1.5 linux kernel

USBmd64 - USB driver kernel module for network analytics, debugging etc., based on 64-bit 4.13.3 linux kernel

ASFER - AStroinFER - BigData Analytics and Machine Learning Software for Large Data Sets

KINGCOBRA - Linux kernelspace messaging with miscellaneous applications based on 32-bit 4.1.5 linux kernel

KINGCOBRA64 Linux kernelspace messaging with miscellaneous applications based on 64-bit 4.13.3 linux kernel

Krishna iResearch DoxygenDocs - Documentation for NeuronRain open source product codebases (NeuronRain - Architecture Diagrams 1 , NeuronRain - Architecture Diagrams 2)

NeuronRain Documentation in ReadTheDocs and FAQ

Krishna iResearch (old link updated in 2006)

Bifurcation of code bases into 32-bit and 64-bit previously has been chosen than branching/ifdefs because of drastic difference in linux kernel versions/functionalities/dependencies between base kernels for 32 and 64 bits. The opensource codebases in SourceForge and GitHub above are non funded,non profit academic research efforts. Premium technical support is available for above opensource codebases. GitHub repositories implement NeuronRain Enterprise and SourceForge repositories implement NeuronRain Research versions. Dual licensed closedsource premium commercial versions with enhanced features on above GPL products in development since 2010.

GRAFIT Open Learning - Free online course materials

Research statements

Research statement 1 (2010)

Research statement 2 - with some proof sketches (2011) (Algorithm for Timeout implemented on Global Decisioning Platform 3.0 - Copyright: Global Analytics)

Research statement 3 - with some proof sketches (2014)

Team Patents (Sun Microsystems - 2000-2005)

Disclosure - Patent Pending (Copyright: Sun Microsystems - 2002)

Survival Index Based Transaction Timeout Manager (Java PoC implemented on SunOne-iPlanet Application Server 6.5 J2EE-JTS Transaction Manager)

Publications (2008-2011) - (CMI-IMSc-IIT, Chennai) guided and reviewed (Advisors/Co-Authors in Google Scholar)

Few Algorithms for Ascertaining Merit Of a Document

(Master's thesis) arXiv Link to Few Algorithms for Ascertaining Merit of a Document

Presentation slides - Few Algorithms for Ascertaining Merit Of a Document

(Published during PhD - October 2010) TAC 2010 dataset evaluation - Update summarization with Interview Algorithm (with some updates added to the above)- paper

TAC 2010 dataset evaluation - Update summarization with Interview Algorithm (with some updates added to the above)- slides

NIST TAC 2010 link to Update Summarization with Interview Algorithm

Decidability of Existence and Construction of a Complement of a given function

arXiv Link to Decidability of Existence and Construction of a Complement of a given function

Circuits for Complement of a function - old version

Publication Drafts (PDF and Text) (NOTE: These are not final versions and are work in progress) (2012 - present) - unguided and unreviewed

Complexity Theoretic Analysis of Non-majority and Majority Social Choice, Pseudorandomness, Goodness of Voting and Condorcet Jury Theorem, Complement Function, Ramsey coloring of sequences, Diophantine Analysis, Riemann Zeta Function, Hypergeometric Functions, Graph theoretic/Computational linguistic/Interview Intrinsic Merit/Fitness and Experiential Learning in the context of WWW (mostly text analysis) and Social networks, Flow Market Equilibrium and Merit Equilibrium, Neural Networks and Deep Learning, Quantum mechanics and Intrinsic Fitness, Hash Functions, Integer Partitions, Space filling/Tiling/Packing, Satisfiability (CNFSAT and QBFSAT), Linear and Convex-Concave Programming, Computational Geometric Integer Factoring and

Connections amongst them etc., ***(most recent draft updates to all publications previously and earlier drafts below - in text format - nonlinear theoretical writeups interspersed between NeuronRain AsFer code commits in SourceForge and GitHub):***

NeuronRain Research AstroInfer Design Document [SourceForge]

NeuronRain Enterprise AstroInfer Design Document [GitHub]

Earlier Drafts:

Integer Partitions and Hash functions (in Tex) - 2012

Interview Algorithm is in IP=PSPACE (in Tex) - 2012

Few Non-trivial Questions and Shell Turing Machines (in Tex) - 2012

Complexity aspects of Non-majority and Majority Social Choice, Hash Functions, Integer Partitions, Graph theoretic Document summarization etc.,:

Arrow’s Theorem, Circuit For Democracy and Pseudorandom Choice and P Versus NP - (Draft - 17 September 2014)

Document Summarization from WordNet Subgraph obtained by Recursive Gloss Overlap (Draft - 25 July 2014)

Integer Partitions and Hash Functions (new version - 5 April 2014 and 17 April 2014)

Lower Bounds for Majority Voting and Pseudorandom choice

Circuits For Computing Error Probability of Majority Voting (new version - 10 April 2014)

Circuits For Computing Error Probability of Majority Voting (old version - March 2013)

In-depth Analysis of a Variant of Majority Voting with relation to ZFC - updated draft (new version - 8 February 2014)

In-depth Analysis of a Variant of Majority Voting with relation to ZFC (old version - 8 March 2013)

Parallel PRG and Space Filling:

A Chaos theoretic Parallel Pseudorandom generator in RNC For Majority Voting and Pseudorandom Choice

Analysis of a Randomized Space Filling Algorithm and its Linear Program Formulation - updated draft additions for Cellular Automaton Algorithm, NC circuit construction for it

Analysis of a Randomized Space Filling Algorithm and its Linear Program Formulation (previous version)

Discrete Hyperbolic Factorization - previous versions:

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization **(Version 1)**

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - with Interpolation Search **(Version 2 - updated 25 June 2013)**

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - with Interpolation Search **(Version 3 - updated 30 June 2013 with** rough notes**)**

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - with Interpolation Search **(version 4 - updated 1 July 2013 and Version 5 - updated 20 July 2013 including all handwritten notes)**

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - using Rectangular Binary (or) Interpolation Search (version 12 - updated 25 August 2013)

Informal Notes on Derivation of Upperbound for Discrete Hyperbolic Factorization with Stirling Formula - using Rectangular Binary or Interpolation Search (10 September 2013)

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - using Rectangular Binary (or) Interpolation Search applying Stirling Formula (Version 14 - 20 September 2013)

Discrete Hyperbolic Factorization - Parallel RAM algorithm:
An NC algorithm and some Sequential Search Algorithms for Discrete Hyperbolic Polylogarithmic Sieve For Factorization using Binary or Interpolation Search with Stirling Formula and Logarithmic Sorted Tile Merge in PRAM model (20 November 2013**) and**
AsFer PRAM implementation design notes with tile id(s) (21 November 2013)

An NC algorithm and some Sequential Search Algorithms for Discrete Hyperbolic Polylogarithmic Sieve For Factorization using Binary or Interpolation Search with Stirling Formula and Logarithmic Sorted Tile Merge in PRAM model - updated draft with PRAM to NC reduction and input size details and references (25 September 2014)

Miscellaneous Informal Notes related to above drafts (Handwritten) (Note: these are not in any structured format and might have typos and errors)

Implication Graphs, Error probability of Majority Voting and P Versus NP Question

Minimum Convex Hulls of Implication Graphs and Hidden Markov Model on class nodes of Concept Hypergraph

Minimum Convex Hulls of Implication Random Growth Networks and Perfect Voter Decidability

Philosophical Analysis of Democracy Circuit and Pseudorandom Choice

Schur's Theorem, Restricted Partitions with distinct parts and Hash Table Collision Chains

Riemann Zeta Function, Ramanujan Graphs and Ihara Zeta Function - (30 August 2014)

Riemann Zeta Function, Ramanujan Graphs and Ihara Zeta Function - (25 October 2014)

Miscellaneous notes on Krishna iResearch Open Source products design, Democracy Circuit, Complement Function circuit and Parallel RAM to NC reduction for ANSV algorithm in Discrete Hyperbolic Factorization - (6 January 2015)

Publication Drafts (TeX) (NOTE: These are not final versions and are work in progress)

Arrow’s Theorem, Circuit For Democracy and Pseudorandom Choice and P Versus NP - (Draft - 17 September 2014)

Document Summarization from WordNet Subgraph obtained by Recursive Gloss Overlap (Draft - 25 July 2014)

Integer Partitions and Hash Functions (new version - 5 April 2014 and 17 April 2014)

Lower Bounds for Majority Voting and Pseudorandom choice

Circuits for Computing Error Probability of Majority Voting (new version - 10 April 2014)

Circuits For Computing Error Probability of Majority Voting (old version - March 2013)

In-depth Analysis of a Variant of Majority Voting with relation to ZFC - updated draft (new version - 8 February 2014)

In-depth Analysis of a Variant of Majority Voting with relation to ZFC (old version - 8 March 2013)

Parallel PRG and Space Filling:

A Chaos theoretic Parallel Pseudorandom generator in RNC For Majority Voting and Pseudorandom Choice

Analysis of a Randomized Space Filling Algorithm and its Linear Program Formulation

Discrete Hyperbolic Factorization - previous versions:

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization (Version 1)

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - with Interpolation Search (Version 2 - updated 25 June 2013)

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - with Interpolation Search (Version 3 - updated 30 June 2013)

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - with Interpolation Search (Version 4 - updated 1 July 2013 and Version 5 - updated 20 July 2013 including all handwritten notes)

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - using Rectangular Binary (or) Interpolation Search (Latest - version 12 - updated 25 August 2013)

Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - using Rectangular Binary (or) Interpolation Search applying Stirling Formula (20 September 2013)

Discrete Hyperbolic Factorization - Parallel RAM algorithm:
An NC algorithm and some Sequential Search Algorithms for Discrete Hyperbolic Polylogarithmic Sieve For Factorization using Binary or Interpolation Search with Stirling Formula and Logarithmic Sorted Tile Merge in PRAM model (20 November 2013)

An NC algorithm and some Sequential Search Algorithms for Discrete Hyperbolic Polylogarithmic Sieve For Factorization using Binary or Interpolation Search with Stirling Formula and Logarithmic Sorted Tile Merge in PRAM model - updated draft with PRAM to NC reduction and input size details and references (25 September 2014)

Blogs

HAMSA (Computer Science - Machine Learning etc., - complements GRAFIT course material)

VARAHAMIHIRA (Astronomy, Sanskrit etc.,)

Personal Memorabilia and selected photos

Passport (May 2015)

At Mahabalipuram - September 2012

View of SIPCOT TCS from CMI in twilight - August 2010

CMI Alumnus page (2010-)

CMI Research Scholar Page (2010-2011)

Sun Microsystems group photo 1 (2000)

Sun Microsystems 2 (2004)

Sun Microsystems 3 (2004)

COBRA (a not-so-naive cloud precursor implemented during BE in 1999 on CORBA)

PSG Tech Photos (1995-1999)

PSG Tech Alumnus Page (1999-)

Past photos - 1 (2003)

Past photos - 2 (2003)

STATUTORY DISCLAIMER:This website contains publications and articles devoted to multi-disciplinary fundamental research only. Any misinterpretation with malafide intent or defacing/hacking or any other form of cybercrime on contents of this website will be reported and severely dealt with as the case may be. Earlier such incidents have already been reported few years ago. Copyright of this website rests with KaShrinivaasan (alias) Shrinivas Kannan (alias) Srinivasan Kannan