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 $1-p$ ($0 <= \mathrm{p} <= 1$). Let $\mathrm{p} = 0.5$ 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. Accord-

ingly, 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, $I = V \cdot E \cdot Rel/f$ 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

$$Rel = \sum_{i=1}^{n} (relatedness(Level(i), keyword1, keyword2, ..., keywordn)) \tag{9}$$

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,

$$relatedness(Level(i), keyword1, keyword2, ..., keywordn) = OverlapLengthAtLevel(i)(LinearOverlap) \tag{10}$$

(or)

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

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