TAC 2010 - Update Summarization applying Interview Algorithm

Ka.Shrinivaasan
Chennai Mathematical Institute
shrinivas@cmi.ac.in

Motivation

- Is prestige based ranking perfect?
- Are there alternatives?
- Two judging traditions – majority voting and interactive – which is right? Subjective or objective?
- Can a document be analyzed independently to get its quality?

Three Algorithms ...

- **Citation Graph Maxflow and Path Lengths (uses Recursive gloss overlap for sentiment analysis - finding polarity)**

- **Definition Graph Convergence (or) Generalized Recursive Gloss Overlap**

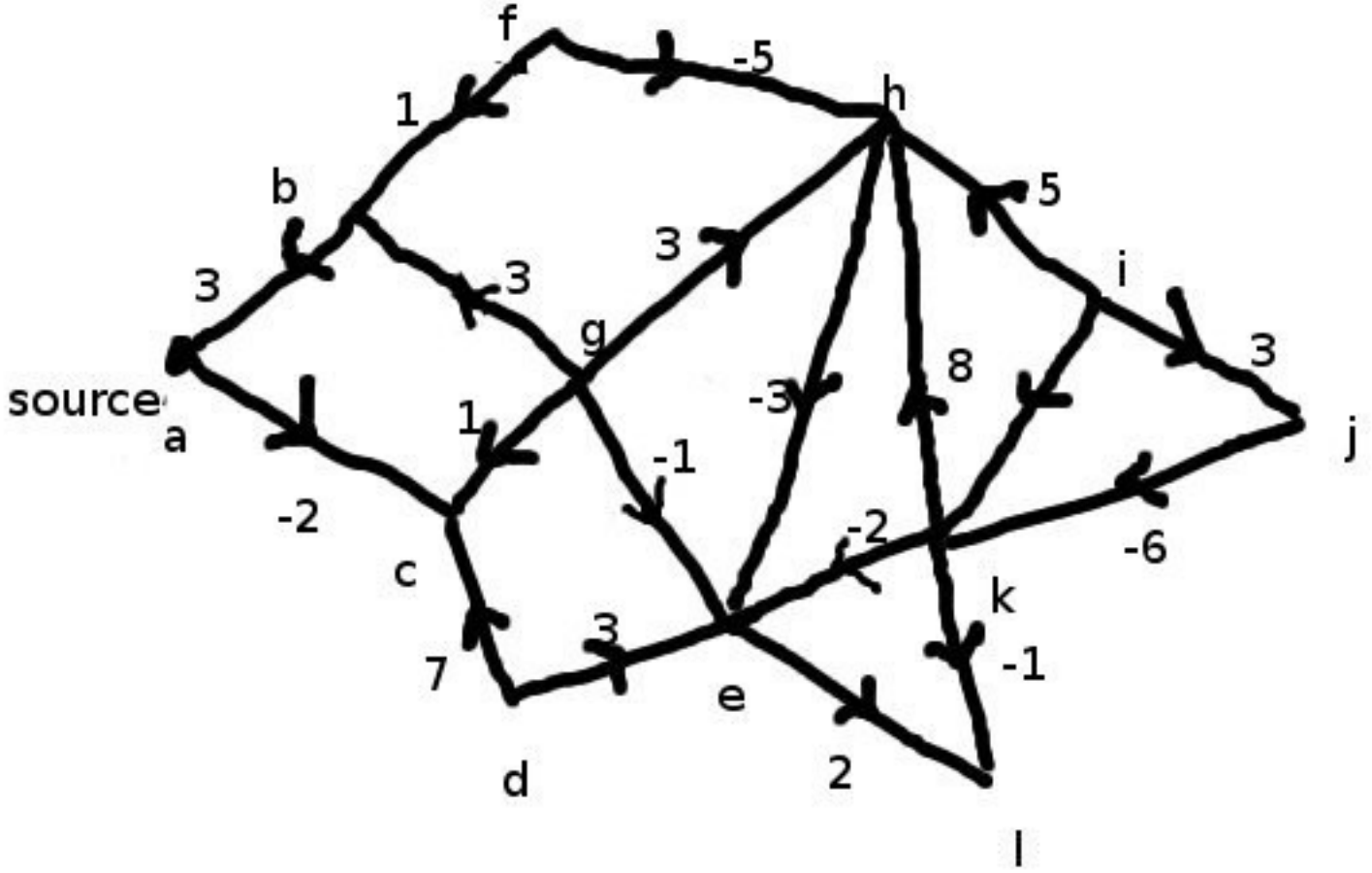
- **Interview algorithm – applies either of above**

- Application to Update summarization

Part I - Directed Graph of Citations

- Merit = influence on future documents = citations
- Construct a directed graph of citations
- Weight of an edge (u,v) = No. Of citations of u by v (is this only way to weight?)
- Polarity of (u,v) = Sentiment Analysis of Citation Context – Positive or Negative
- Number of nodes in all paths of fixed length from source s is a measure of merit (might mislead)

Citation digraph - How it looks



Mincut/Maxflow of Citation DiGraph

- Get Maxflow from Ford-Fulkerson algorithm with each distinct vertex pair as (source, sink) – Capacity of the edge = weight of the edge

- Mincut of citation graph carries Maximum Flow of the concept from source document s - “most influenced by the source document s”

- Average Maxflow out of a source s, is thus a measure of merit of s (= (∑mxf(s,t)) / (|V|-1))

Part II – Definition Graphs

- “Fruit”

- Evocative - What do we get reminded of after reading the above? *(plant, tree, sweet, taste, food, juice, result ...?)*

- Evocation WordNet

Human thought process and Definition Graphs

- Humans scan through the natural language text
- Relate the keywords – motivation behind WordNet
- **What distinguishes the merit of 2 documents X and Y? Grammatical correctness? *No*. Both X and Y written equally grammatically. Content and Complexity? *Yes*. How to measure?**

Recursive Understanding - An Example

- *Document: " Car race ends with flag waving"*

- What is "Car"? Car is an automobile
  - What is "automobile"? Fuel driven Machine
    - What is "Fuel"? Petroleum ...
- What is "race"? Race is ethnic group; contest
  - What is "contest"? Game
    - What is "Game" ? Play ...
- What is "end"? ...
- What is "flag"? ...
- What is "waving"? ...

Definition Graph Convergence (or) Generalized Recursive Gloss Overlap

- **<u>Meaningfulness:</u> “Meaningful” text has its keywords' Synsets within threshold WordNet distance (e.g Jiang-Conrath)**

- WordNet relates words by relations - “is-a”, “has-a” etc., - SYNonymous SETs

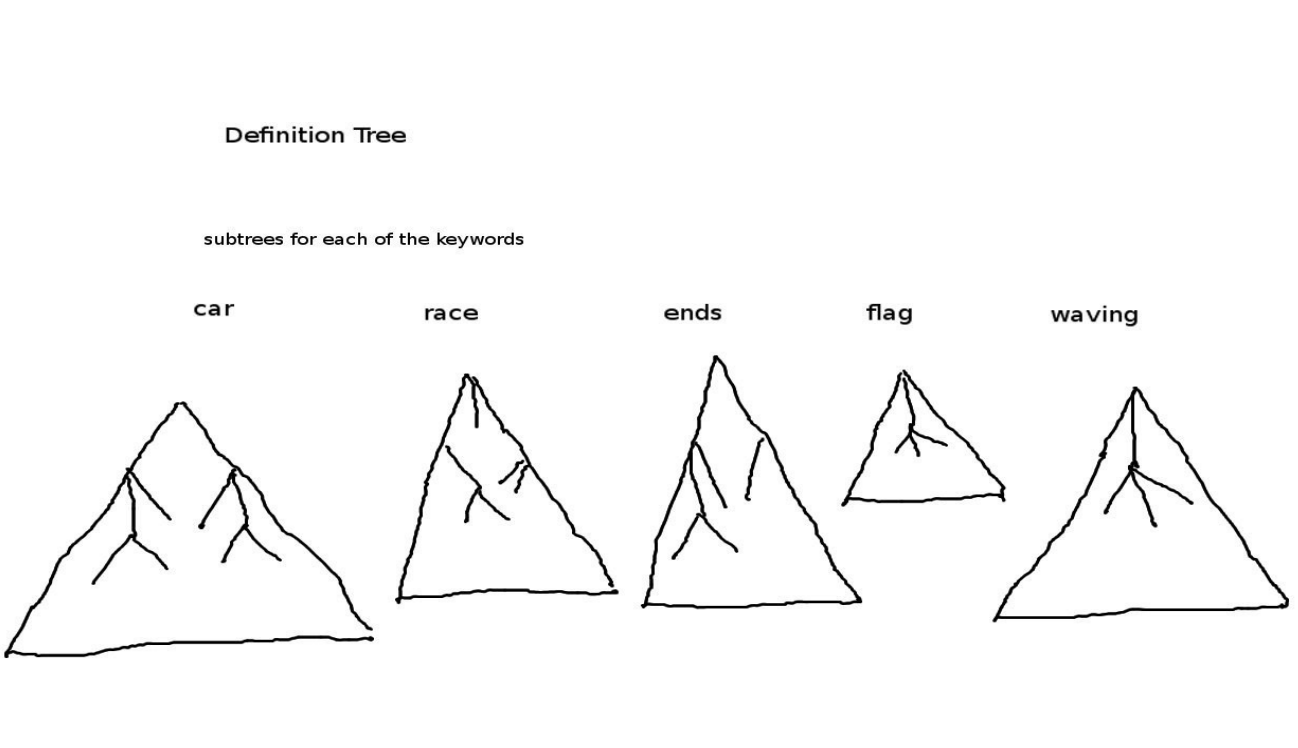
- *Map a document to a **subgraph** of WordNet (Definition Trees/Graphs): F(Document) = G(V,E)*

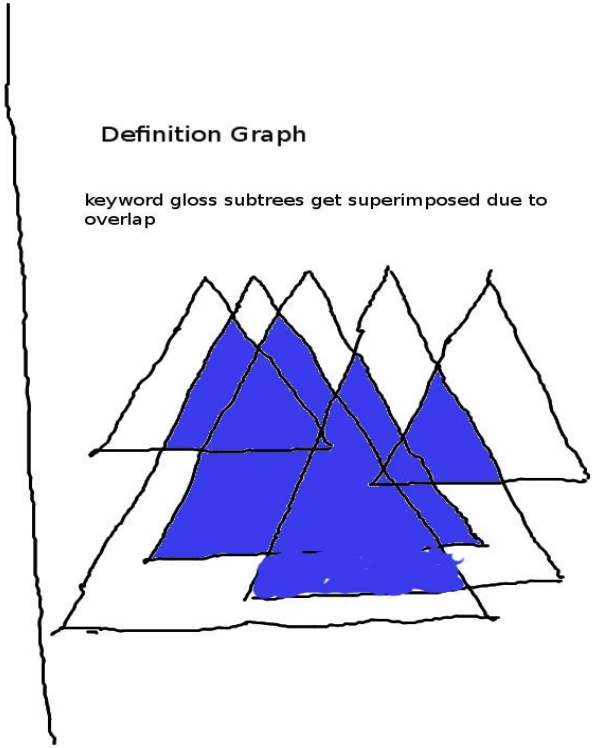
Definition tree and Definition graph

- *DefinitionTree(keyword) = DefinitionTree(gkeyword1) DefinitionTree(gkeyword2)DefinitionTree(gkeyword3) ... DefinitionTree(gkeywordn)* where gkeyword1 through gkeywordn are in the gloss(keyword)

- N subtrees obtained above overlap to form a graph

Definition Tree and Graph - example





Properties of definition graph

- Definition graph is **multipartite**
- Difference in number of vertices in definition tree and definition graph = convergence factor
- Convergence factor is due to gloss overlap - indicator of relatedness
- Relatedness differentiates 2 documents
- We do not consider grammatical structure

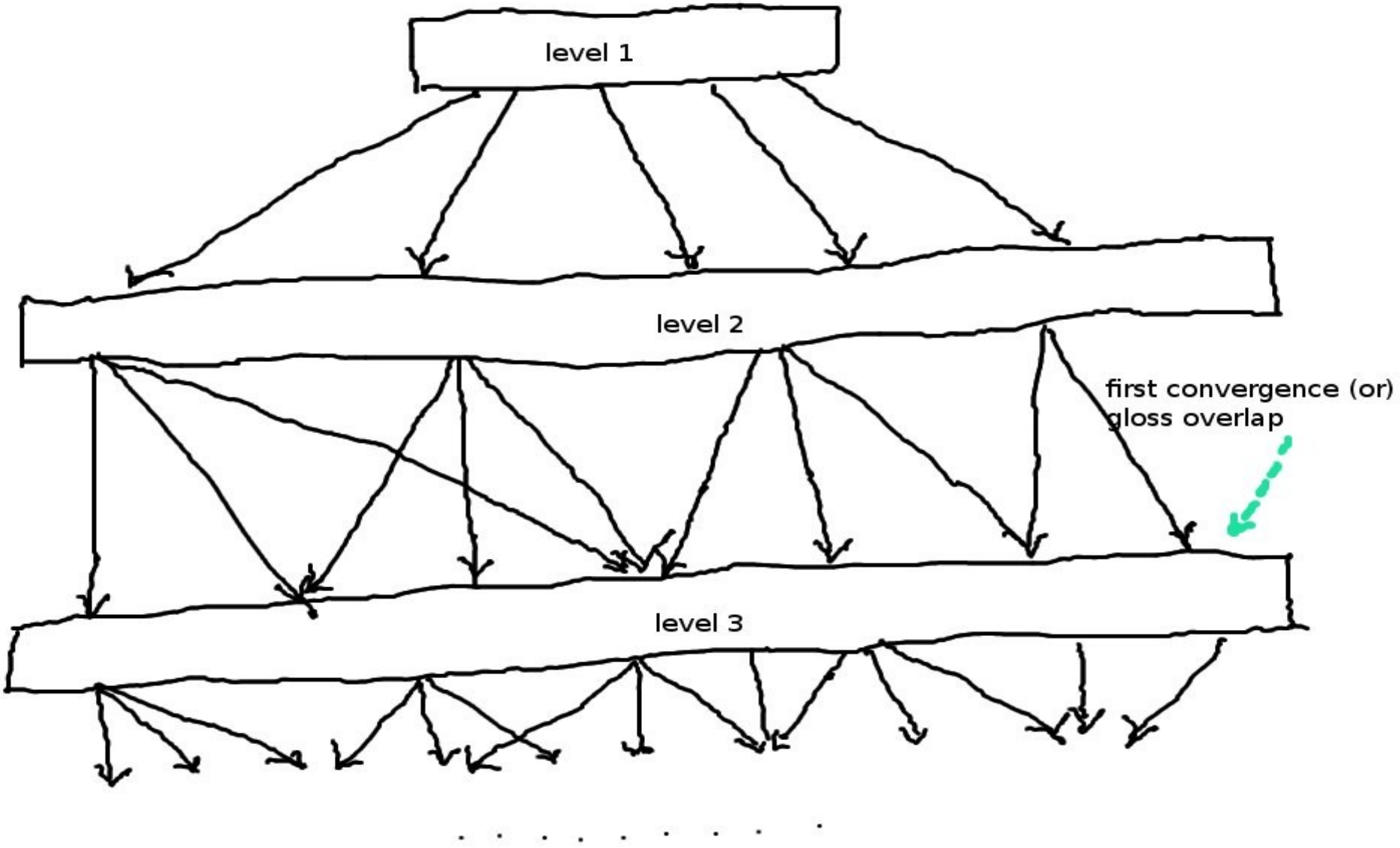
Properties of Definition Graph(contd...)

- Multipartiteness - vertices are partitioned into sets; edges only amongst the sets – useful for preserving recursion level and multipartite-cliques

- Degrees of vertices can be thought of as “votes” for a “theme” keyword – **unsupervised text classifier**

- Context-sensitiveness still present - Word Sense Disambiguation is done during graph construction

Definition Multipartite Graph Visualised



Recursive Gloss Overlap algorithm

1) Get the document as input

2) keywordsatthislevel = {keywords from the document through tf-idf filter (implementation uses 0.02)}

3) While (current_level < depth_required) {

- For each keyword from keywordsatthislevel lookup the **best matching definition(WSD) for the keyword** and add to a set of tokens in next level

Recursive Gloss Overlap algorithm(contd...)

- Remove common tokens (isomorphic nodes) with previous levels - an optimization

- Update the number of vertices*(unique tokens)*, edges*((x,y)='y is in definition of x')* and relatedness *(linear overlap or quadratic overlap)*

- Update keywordsatthislevel

Recursive Gloss Overlap algorithm(contd...)

} //end while

5) Output the ***Intrinsic merit score = |vertices|*|edges|*|relatedness| / first_convergence_level***

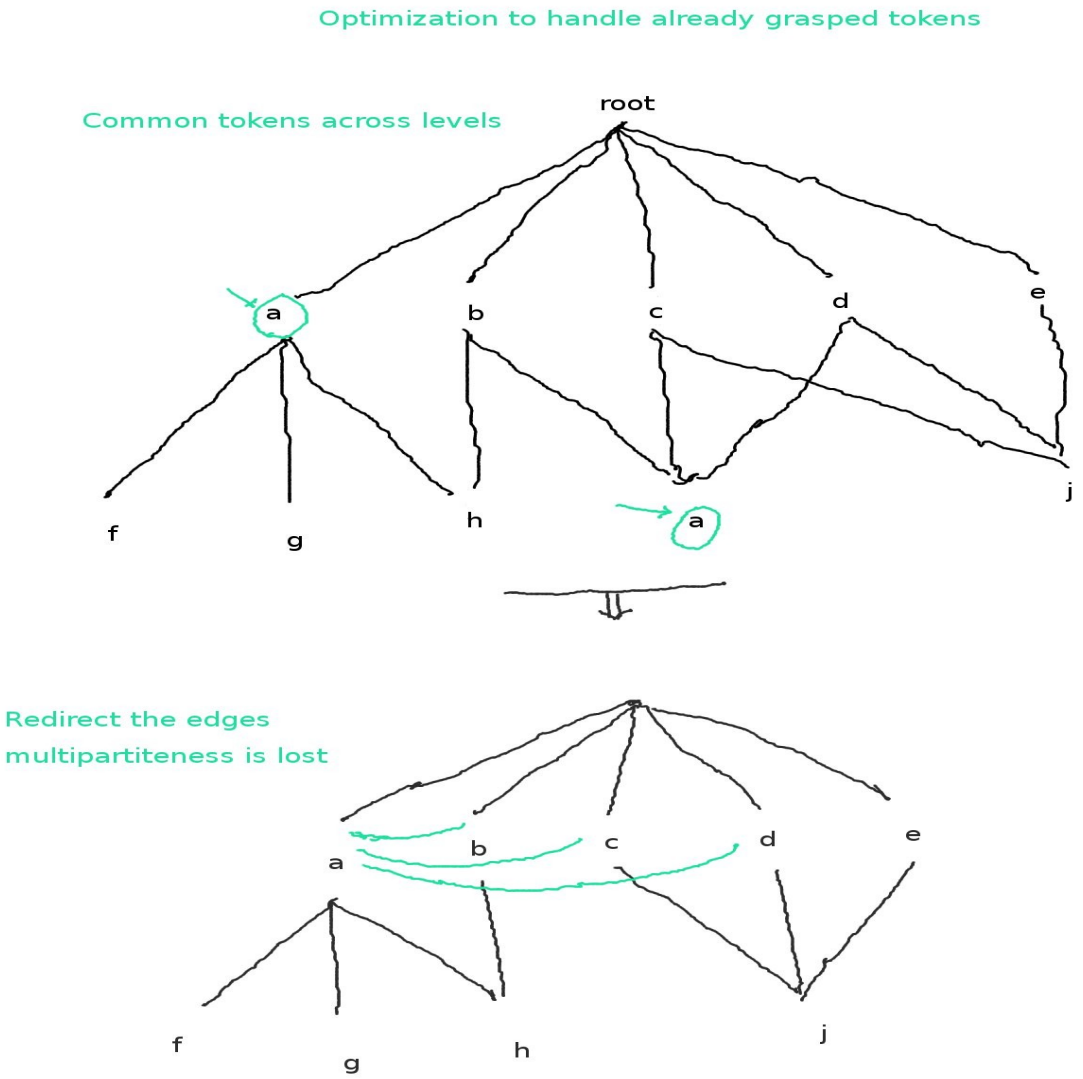
Where

*a) Relatedness = **number of overlaps** (linear, also called as convergence factor) (or)*

*Relatedness = **number of overlapping parents * number of overlaps**2** (quadratic)*

b) First_convergence_level = level of first gloss overlap

Snapshot of Definition graph

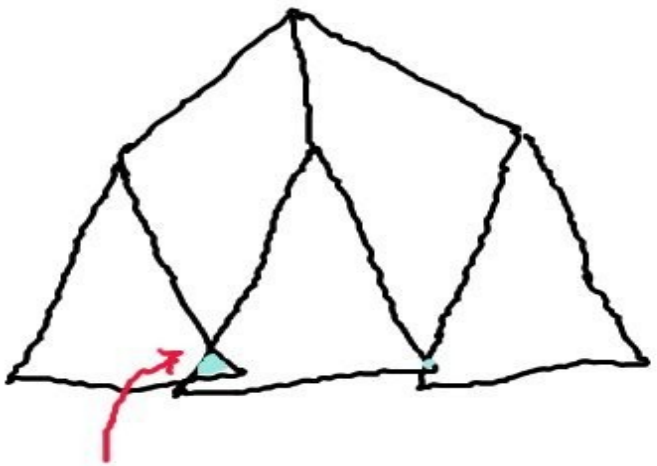


Intuition behind the intrinsic merit score

- vertices ~ knowledge represented by document
- edges ~ relationship among keywords (relation: 'x is in definition of y')
- relatedness ~ complexity quantified by overlap
- first_convergence_level ~ Mingling of definition subtrees
- Above suffice to quantify “meaningfulness” defined earlier *(proportional to V*E*R/ f)*

Comparing two documents for merit

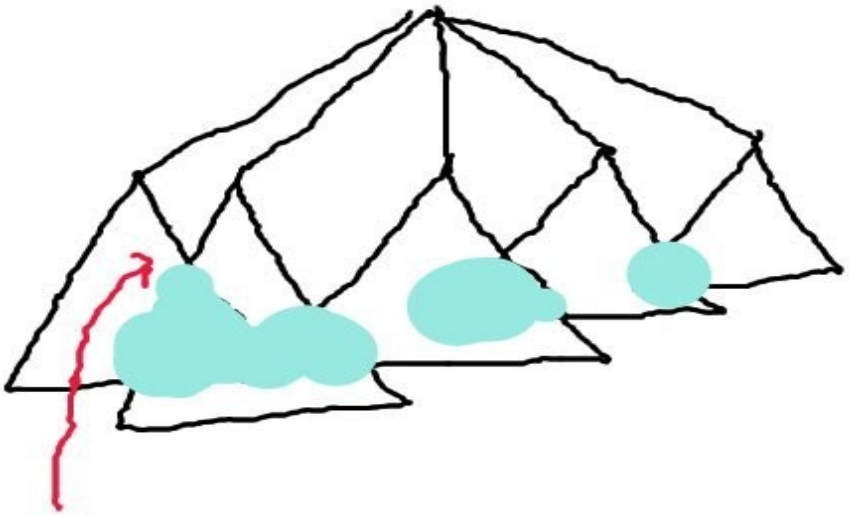
Document1 has less overlap



first convergence level = 5

Example: Car plies on sky

Document2 has more overlap



first convergence level = 2

Example: Cars and buses ply on road

BFS/DFS of definition graph

- Visiting all nodes of definition graph – O(V+E)
- But this does not take into account the relatedness
- Worst case size of definition graph is O(x□ d) *(where x is the average size of a keyword definition and d is the depth)*
- For a meaningful document, overlaps bring down this to great extent – no exponential blowup- O(V)

Pros and Cons

- No False negatives, False positives exist

- Other ranking schemes can be derived from definition graph – based on graph connectedness, completeness etc.,

- Definition graph relies on relatedness of words instead of hyperlink graph – one more layer of abstraction – **all documents now become projected subgraphs of larger universal semantic graph (e.g WordNet)**

Running time of Recursive Gloss Overlap Algorithm

Worst case running time of Recursive gloss overlap is O(E*(V^2)) where V is the number of vertices and E the edges of graph

This is sum total of the running times for gloss lookups, merging duplicate vertices and isomorphic nodes removal.

Application of Recursive gloss overlap to sentiment analysis

- Needed - SentiWordNet - gloss with quantified positivity/negativity score for a keyword

- Example: *"That movie was fantastic. Graphics was awesome"*

- Def Graph level 1: {movie:*motion picture;+0.1*, fantastic:*great;+0.7*, graphics:*software technique; +0.05*, awesome:*great;+0.7*}

- Polarity of Overlap – {great} with positivity score +0.7

Parallelizability of Recursive Gloss Overlap

- Def Graph construction parallelizable - set of tokens of each level broken into subsets
- Assign each subset to a processor (Map)
- Get the results of gloss lookup for subsets and merge them (Reduce)
- To do – Apply MapReduce framework to Recursive Gloss Overlap – E.g Needs a Hadoop cluster

Part III – Interview Algorithm

- Reference "interviews" the candidate – both are documents

- Candidate is inducted into reference if the interview score is above threshold

- Interview is less invasive compared to definition graph construction

- Tree/Graph of interviews can be built (transitive) –e.g x interviews (y,z), y interviews w, z interviews p

Interview Algorithm (contd...)

- Intrinsic merit of candidate measured by either a) Citation Digraph or b) Recursive gloss overlap algorithms

- Interview - a) **supervised** (reference Q&A available) or b) **unsupervised** (reference Q&A are computed from reference – 'Q's are keywords / 'A's are contexts)

- Interview is the set of tuples = {t(1), t(2), ...,t(n)}
  t(i) = (question,answer,expected_answer,score)

Interview Algorithm (contd...)

- Total interview score = ∑(t(i).score) ***(where t(i).score = |shingles(answer) ∩ shingles(expected_answer)| / |shingles(answer) U shingles(expected_answer)|***

- Value addition = edit distance of DefGraph(Reference) and DefGraph(Candidate) ***(where EditDistance(G,H) = |edges added| + |edges removed| to transform G into H)***

- Final score = ***w1 * intrinsic_merit + w2 * interview_Q&A_score + w3 * value_addition, where w1,w2 and w3 are weights***

Application of Interview Algorithm to TAC 2010 Update summarization

- Split each dataset into candidate and reference

- Go through the Interview algorithm and get scores for candidates (for intrinsic merit, recursive gloss overlap was used)

- Choose the best candidate and update the summary after sentence scoring (sentence score = sum of tf-idf values of constituent terms)

[NOTE: Only 25 (out of 92) datasets were tried due to hardware issue]