Text Mining for Social Media

CIS 700/CSE 791

Week 13: Text Summarization

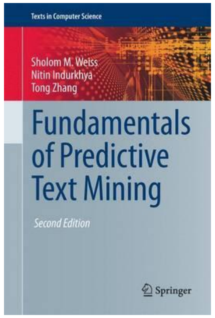
`

Edmund Yu, PhD

Associate Professor

esyu@syr.edu

April 12, 14, 2016



Texts in Computer Science

Sholom M. Weiss
Nitin Indurkhya
Tong Zhang

Fundamentals of Predictive Text Mining

Second Edition

Springer

Text Summarization

- A principal technical approach to summarization is closely aligned with **clustering**.
    - A cluster consists of similar documents, which may be considered as having the same topic.
    - We could envision several situations where a summary might be useful:
        - Single document
        - Multiple documents with the same topic
        - An automatically assembled cluster of documents.

Text Summarization

- The common theme is that one or more documents are presented, and from these documents a summary is produced.
    - Don’t let your imagination run wild thinking that a program must rewrite the originals.

      → **Abstraction-based**
    - Our summary will be word-for-word sentences extracted from the given documents.

      → **Extraction-based**

Types of Text Summarization

- **Extraction-based summarization**
  - In this summarization task, the automatic system extracts objects from the entire collection, without modifying the objects themselves.
  - Examples of this include:
    - **Keyphrase extraction**, where the goal is to select individual words or phrases to "tag" a document
    - **Document summarization**, where the goal is to select whole sentences (without modifying them) to create a short paragraph summary.
    - Similarly, in image collection summarization, the system extracts images from the collection without modifying the images themselves.

https://en.wikipedia.org/wiki/Automatic_summarization

Types of Text Summarization

- **Abstraction-based summarization**
  - Abstraction involves paraphrasing sections of the source document.
  - In general, abstraction can condense a text more strongly than extraction, but the programs that can do this are harder to develop as they require use of **natural language generation** technology.
  - While some work has been done in abstractive summarization (creating an abstract synopsis like that of a human), the majority of summarization systems are extractive (selecting a subset of sentences to place in a summary).

https://en.wikipedia.org/wiki/Automatic_summarization

Generating Text from Templates

MESSAGE: ID	TST3-MUC4-0010
MESSAGE: TEMPLATE	2
INCIDENT: DATE	**30 OCT 89**
INCIDENT: LOCATION	EL SALVADOR
INCIDENT: TYPE	**ATTACK**
INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED
INCIDENT: INSTRUMENT ID	
INCIDENT: INSTRUMENT TYPE	
PERP: INCIDENT CATEGORY	TERRORIST ACT
PERP: INDIVIDUAL ID	"TERRORIST"
PERP: ORGANIZATION ID	"THE FMLN"
PERP: ORG. CONFIDENCE	**REPORTED: "THE FMLN"**
PHYS TGT: ID	
PHYS TGT: TYPE	
PHYS TGT: NUMBER	
PHYS TGT: FOREIGN NATION	
PHYS TGT: EFFECT OF INCIDENT	
PHYS TGT: TOTAL NUMBER	
HUM TGT: NAME	
HUM TGT: DESCRIPTION	"1 CIVILIAN"
HUM TGT: TYPE	**CIVILIAN**: "1 CIVILIAN"
HUM TGT: NUMBER	**1**: "1 CIVILIAN"
HUM TGT: FOREIGN NATION	
HUM TGT: EFFECT OF INCIDENT	DEATH: "1 CIVILIAN"
HUM TGT: TOTAL NUMBER	

On October 30, 1989, one civilian was killed in a reported FMLN attack in El Salvador.

Radev and McKeown, 1998, Generating natural language summaries from multiple on-line sources, Computational Linguistics, Volume 24 Issue 3, September 1998 Pages 470-500

Excerpts from 4 Articles

1	JERUSALEM - A Muslim suicide bomber blew apart 18 people on a Jerusalem bus and wounded 10 in a mirror-image of an attack one week ago. The carnage could rob Israel's Prime Minister Shimon Peres of the May 29 election victory he needs to pursue Middle East peacemaking. Peres declared all-out war on Hamas but his tough talk did little to impress stunned residents of Jerusalem who said the election would turn on the issue of personal security.
2	JERUSALEM - A bomb at a busy Tel Aviv shopping mall killed at least 10 people and wounded 30, Israel radio said quoting police. Army radio said the blast was apparently caused by a suicide bomber. Police said there were many wounded.
3	A bomb blast ripped through the commercial heart of Tel Aviv Monday, killing at least 13 people and wounding more than 100. Israeli police say an Islamic suicide bomber blew himself up outside a crowded shopping mall. It was the fourth deadly bombing in Israel in nine days. The Islamic fundamentalist group Hamas claimed responsibility for the attacks, which have killed at least 54 people. Hamas is intent on stopping the Middle East peace process. President Clinton joined the voices of international condemnation after the latest attack. He said the ``forces of terror shall not triumph" over peacemaking efforts.
4	TEL AVIV (Reuter) - A Muslim suicide bomber killed at least 12 people and wounded 105, including children, outside a crowded Tel Aviv shopping mall Monday, police said. Sunday, a Hamas suicide bomber killed 18 people on a Jerusalem bus. Hamas has now killed at least 56 people in four attacks in nine days. The windows of stores lining both sides of Dizengoff Street were shattered, the charred skeletons of cars lay in the street, the sidewalks were strewn with blood. The last attack on Dizengoff was in October 1994 when a Hamas suicide bomber killed 22 people on a bus.

Four Templates

1

MESSAGE: ID	TST-REU-0001
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 3, 1996 11:30**
PRIMSOURCE: SOURCE	
INCIDENT: DATE	**March 3, 1996**
INCIDENT: LOCATION	**Jerusalem**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**“killed: 18"** **“wounded: 10”**
PERP: ORGANIZATION ID	

2

MESSAGE: ID	TST-REU-0002
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 4, 1996 07:20**
PRIMSOURCE: SOURCE	**Israel Radio**
INCIDENT: DATE	**March 4, 1996**
INCIDENT: LOCATION	**Tel Aviv**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**“killed: at least 10"** **“wounded: more than 100”**
PERP: ORGANIZATION ID	

3

MESSAGE: ID	TST-REU-0003
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 4, 1996 14:20**
PRIMSOURCE: SOURCE	
INCIDENT: DATE	**March 4, 1996**
INCIDENT: LOCATION	**Tel Aviv**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**“killed: at least 13"** **“wounded: more than 100”**
PERP: ORGANIZATION ID	**“Hamas”**

4

MESSAGE: ID	TST-REU-0004
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 4, 1996 14:30**
PRIMSOURCE: SOURCE	
INCIDENT: DATE	**March 4, 1996**
INCIDENT: LOCATION	**Tel Aviv**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**“killed: at least 12"** **“wounded: 105”**
PERP: ORGANIZATION ID	

Fluent Summary with Comparisons

Reuters reported that 18 people were killed on *Sunday* in a bombing in Jerusalem. *The next day*, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel radio. Reuters reported that *at least 12 people* were killed and *105* wounded *in the second incident*. *Later the same day*, Reuters reported that Hamas has claimed responsibility for the act.

Operators

- If there are two templates

  AND

  the location is the same

  AND

  the time of the second template is after the time of the first template

  AND

  the source of the first template is different from the source of the second template

  AND

  at least one slot differs

  THEN

  combine the templates using the contradiction operator...

Operators: Contradiction

Contradiction

Precondition:
Different sources report contradictory values for a small number of slots

> The afternoon of February 26, 1993, Reuters reported that a suspected bomb killed *at least six people* in the World Trade Center. *However*, Associated Press announced that *exactly five people* were killed in the blast.

Operators: Change of Perspective

Change of perspective

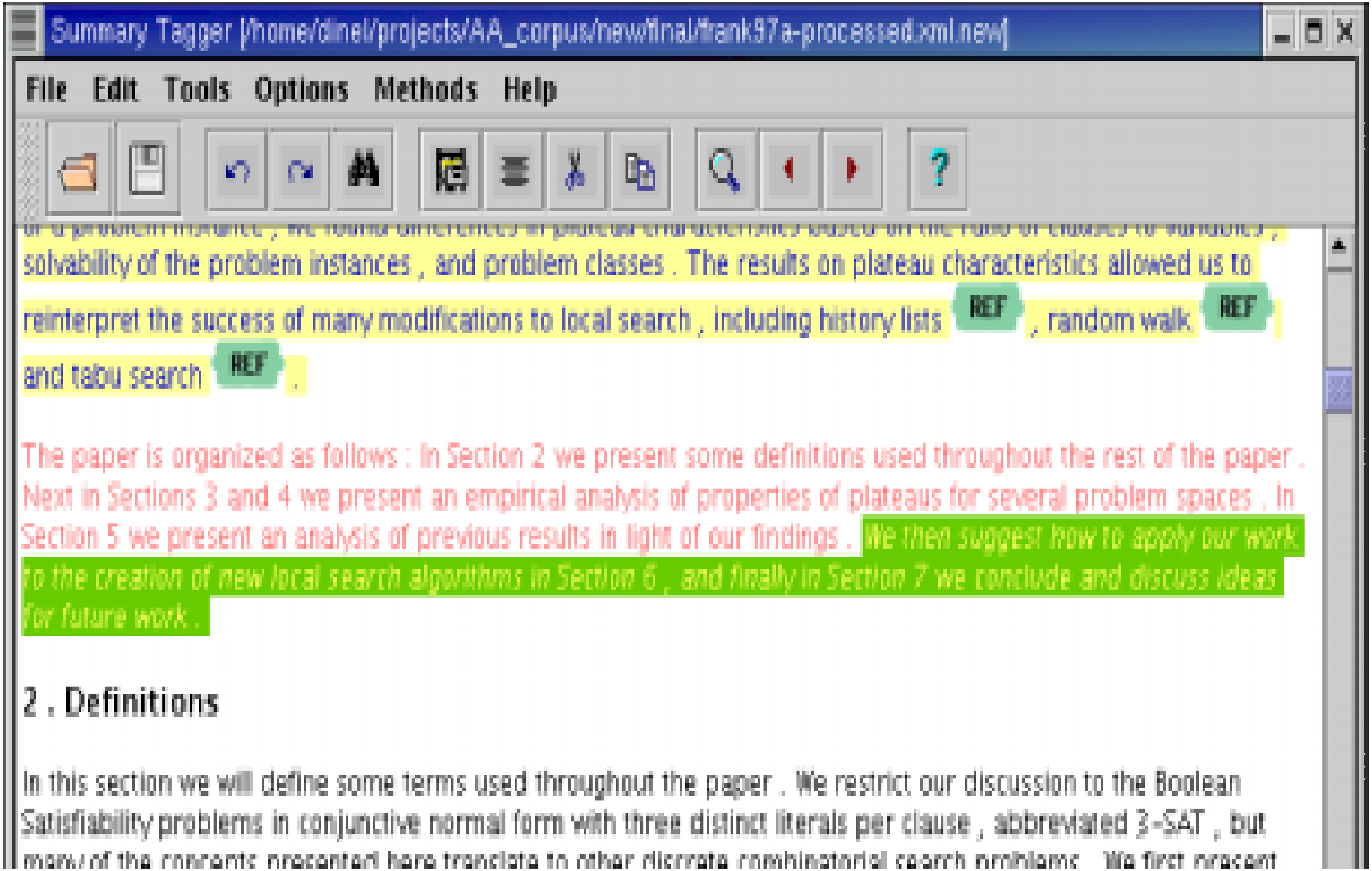
Precondition:
The same source reports a change in a small number of slots

March 4th, Reuters reported that a bomb in Tel Aviv killed at least 10 people and wounded 30. *Later the same day*, Reuters reported that *exactly 12 people* were *actually* killed and *105* wounded.

Types of Text Summarization

- **Aided summarization**
  - Apart from Fully Automated Summarizers (FAS), there are systems that aid users with the task of summarization (MAHS = Machine Aided Human Summarization)
    - For example by highlighting candidate passages to be included in the summary (see next slide)
  - There are also systems that depend on post-processing by a human (HAMS = Human Aided Machine Summarization).

https://en.wikipedia.org/wiki/Automatic_summarization


Summary Tagger [/home/dinel/projects/AA_corpus/newfinal/frank97a-processed.xml.new]
File
Edit
Tools
Options
Methods
Help

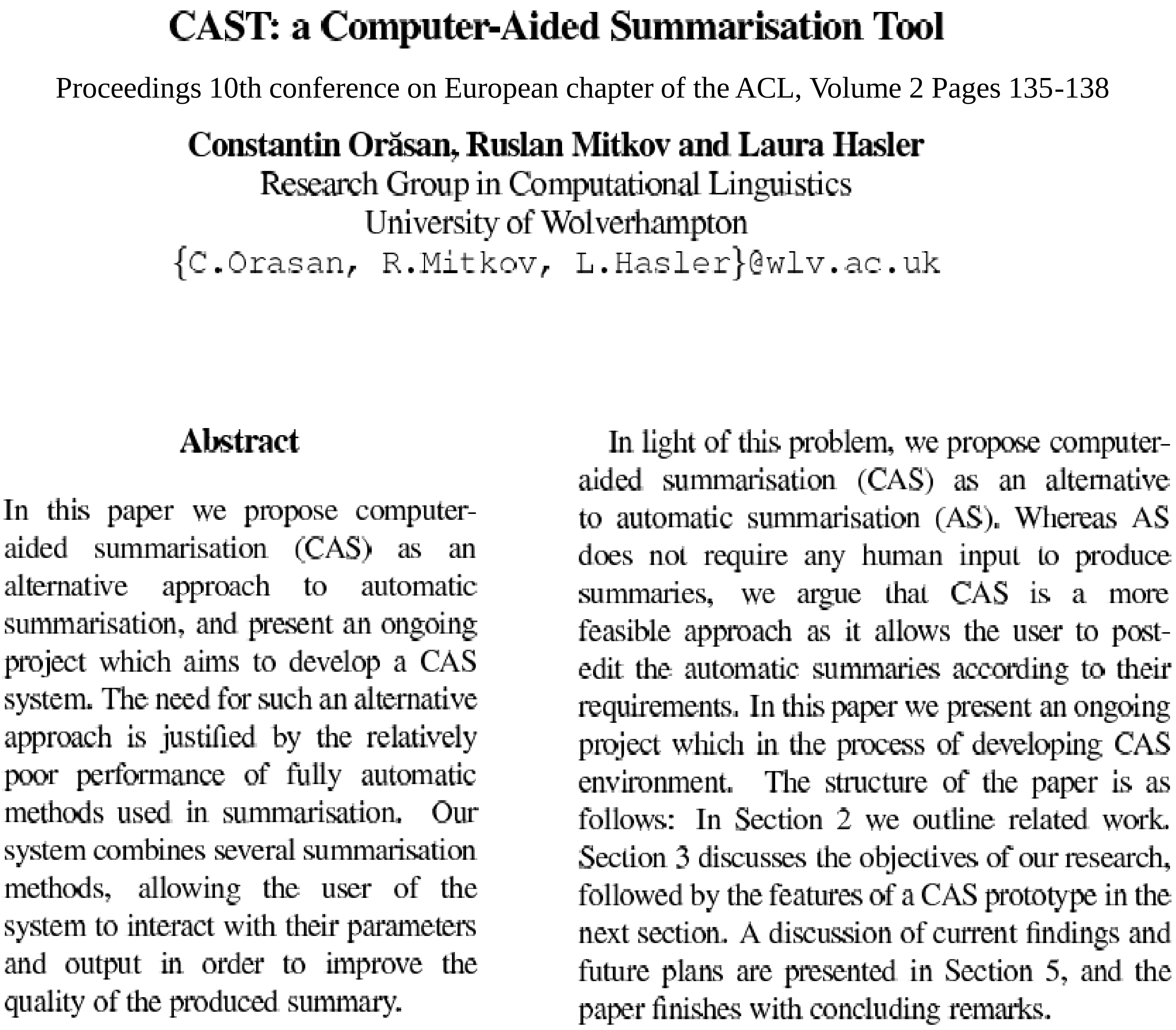
of a problem instance , we found differences in plateau characteristics based on the ratio of clauses to variables , solvability of the problem instances , and problem classes . The results on plateau characteristics allowed us to reinterpret the success of many modifications to local search , including history lists REF , random walk REF and tabu search REF .

The paper is organized as follows : In Section 2 we present some definitions used throughout the rest of the paper . Next in Sections 3 and 4 we present an empirical analysis of properties of plateaus for several problem spaces . In Section 5 we present an analysis of previous results in light of our findings . *We then suggest how to apply our work to the creation of new local search algorithms in Section 6 , and finally in Section 7 we conclude and discuss ideas for future work .*

2 . Definitions

In this section we will define some terms used throughout the paper . We restrict our discussion to the Boolean Satisfiability problems in conjunctive normal form with three distinct literals per clause , abbreviated 3-SAT , but many of the concepts presented here translate to other discrete combinatorial search problems . We first present

Figure 1: Part of the main screen of the tool



CAST: a Computer-Aided Summarisation Tool

Proceedings 10th conference on European chapter of the ACL, Volume 2 Pages 135-138

Constantin Orăsan, Ruslan Mitkov and Laura Hasler
Research Group in Computational Linguistics
University of Wolverhampton
{C.Orasan, R.Mitkov, L.Hasler}@wlv.ac.uk

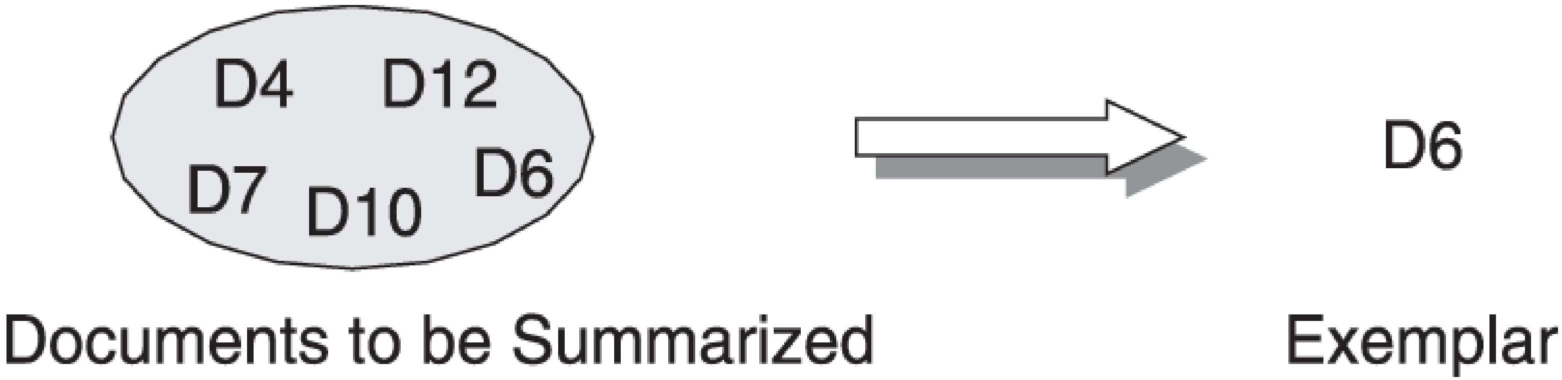
Abstract

In this paper we propose computer-aided summarisation (CAS) as an alternative approach to automatic summarisation, and present an ongoing project which aims to develop a CAS system. The need for such an alternative approach is justified by the relatively poor performance of fully automatic methods used in summarisation. Our system combines several summarisation methods, allowing the user of the system to interact with their parameters and output in order to improve the quality of the produced summary.

In light of this problem, we propose computer-aided summarisation (CAS) as an alternative to automatic summarisation (AS). Whereas AS does not require any human input to produce summaries, we argue that CAS is a more feasible approach as it allows the user to post-edit the automatic summaries according to their requirements. In this paper we present an ongoing project which in the process of developing CAS environment. The structure of the paper is as follows: In Section 2 we outline related work. Section 3 discusses the objectives of our research, followed by the features of a CAS prototype in the next section. A discussion of current findings and future plans are presented in Section 5, and the paper finishes with concluding remarks.

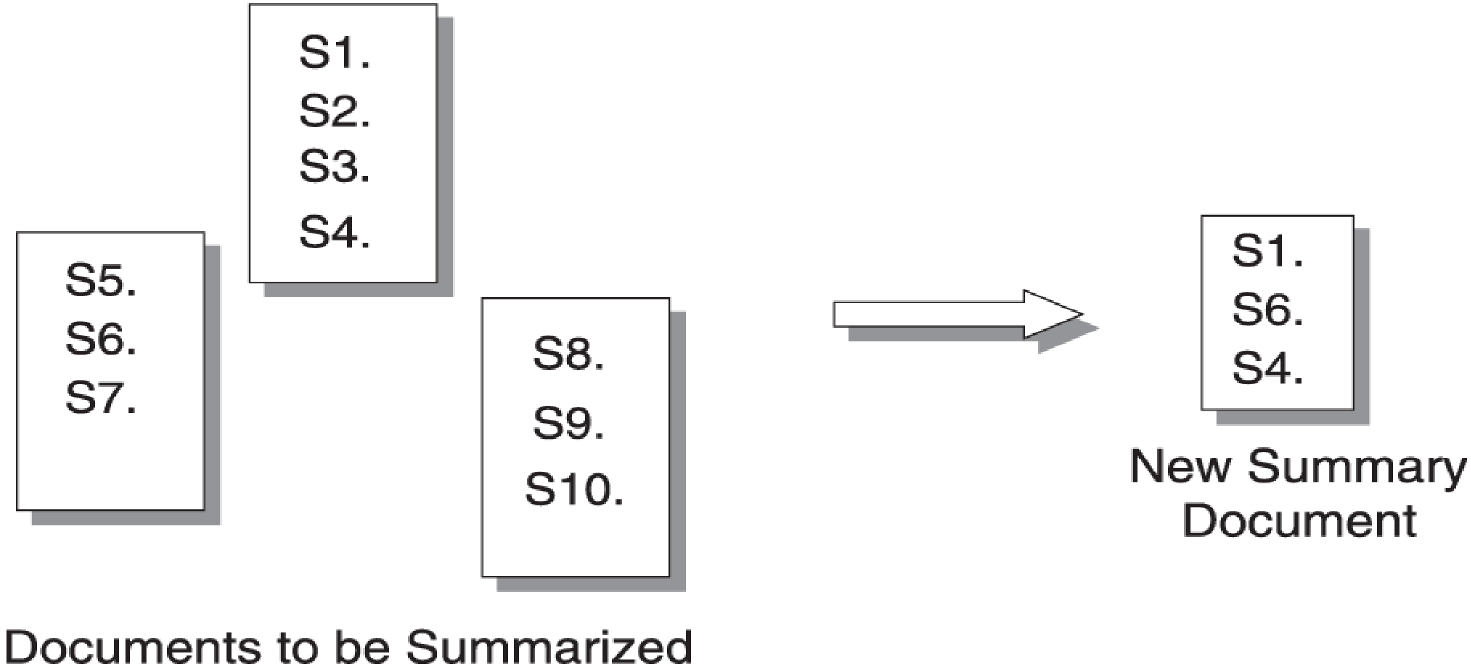
(Extraction-based) Text Summarization

- Our task here is to select the right sentences.
  - Figure 9.1 (below) describes that situation, where a cluster is summarized by one (or more) of its constituent documents.
  - The extract is selected to be representative of the cluster, a summary of the shared topic.



Text Summarization

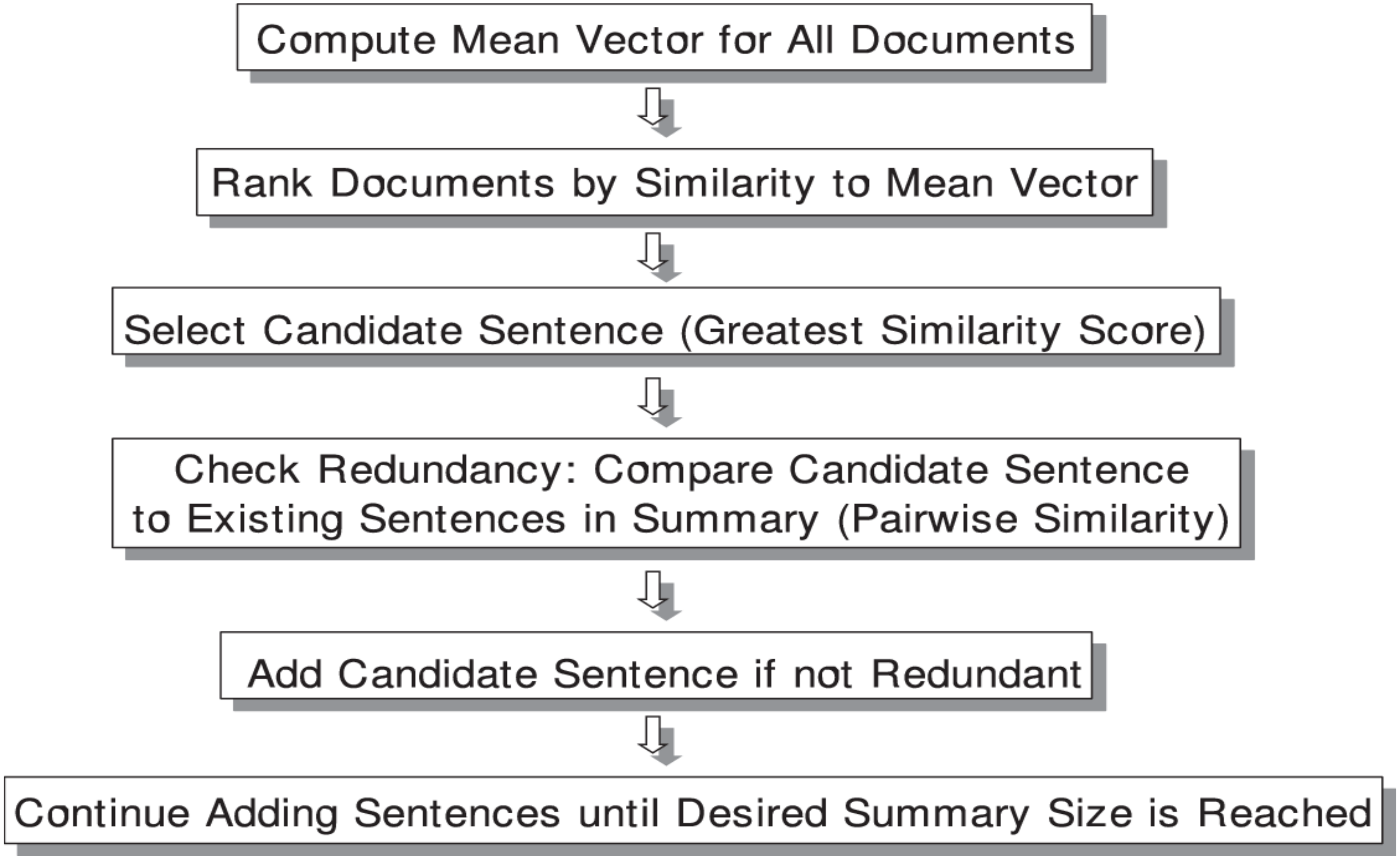
- Figure 9.2 (below) illustrates the extended task of producing a summary by selecting sentences from the given documents
- Instead of selecting a single representative document, the summarizing program extracts sentences from the documents, merging them into a single summary.



Text Summarization

- **How do we select those documents and sentences?**
  - We can use a procedure similar to selecting the document that is most similar to the virtual document represented by the cluster **centroid**.
  - Figure 9.3 (next slide) gives the steps for such a summarization method. (**The centroid method**)

Text Summarization



Compute Mean Vector for All Documents

Rank Documents by Similarity to Mean Vector

Select Candidate Sentence (Greatest Similarity Score)

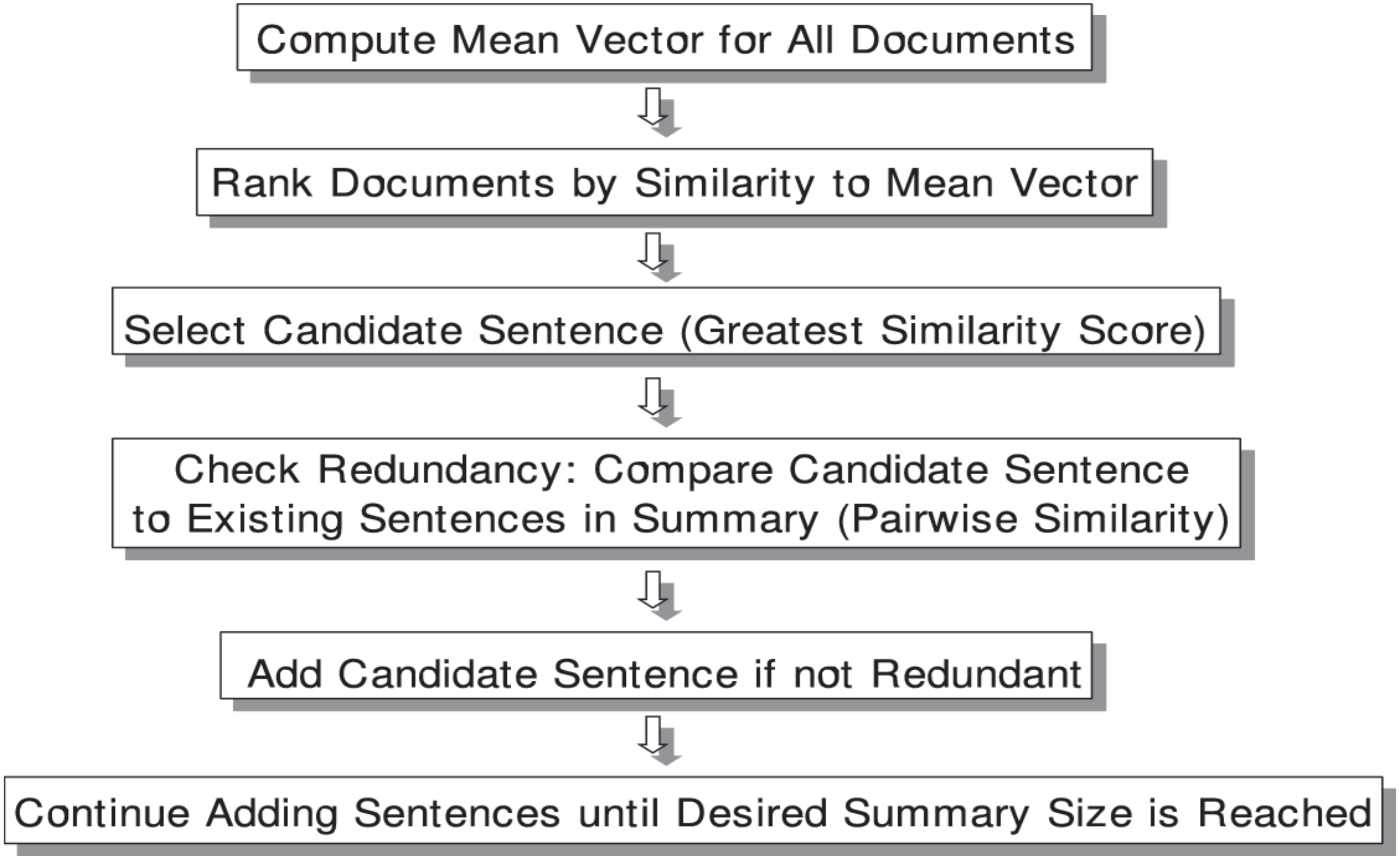
Check Redundancy: Compare Candidate Sentence to Existing Sentences in Summary (Pairwise Similarity)

Add Candidate Sentence if not Redundant

Continue Adding Sentences until Desired Summary Size is Reached

- These sentences are not modified and maintain their relative positions within the original document.

Text Summarization



Compute Mean Vector for All Documents

Rank Documents by Similarity to Mean Vector

Select Candidate Sentence (Greatest Similarity Score)

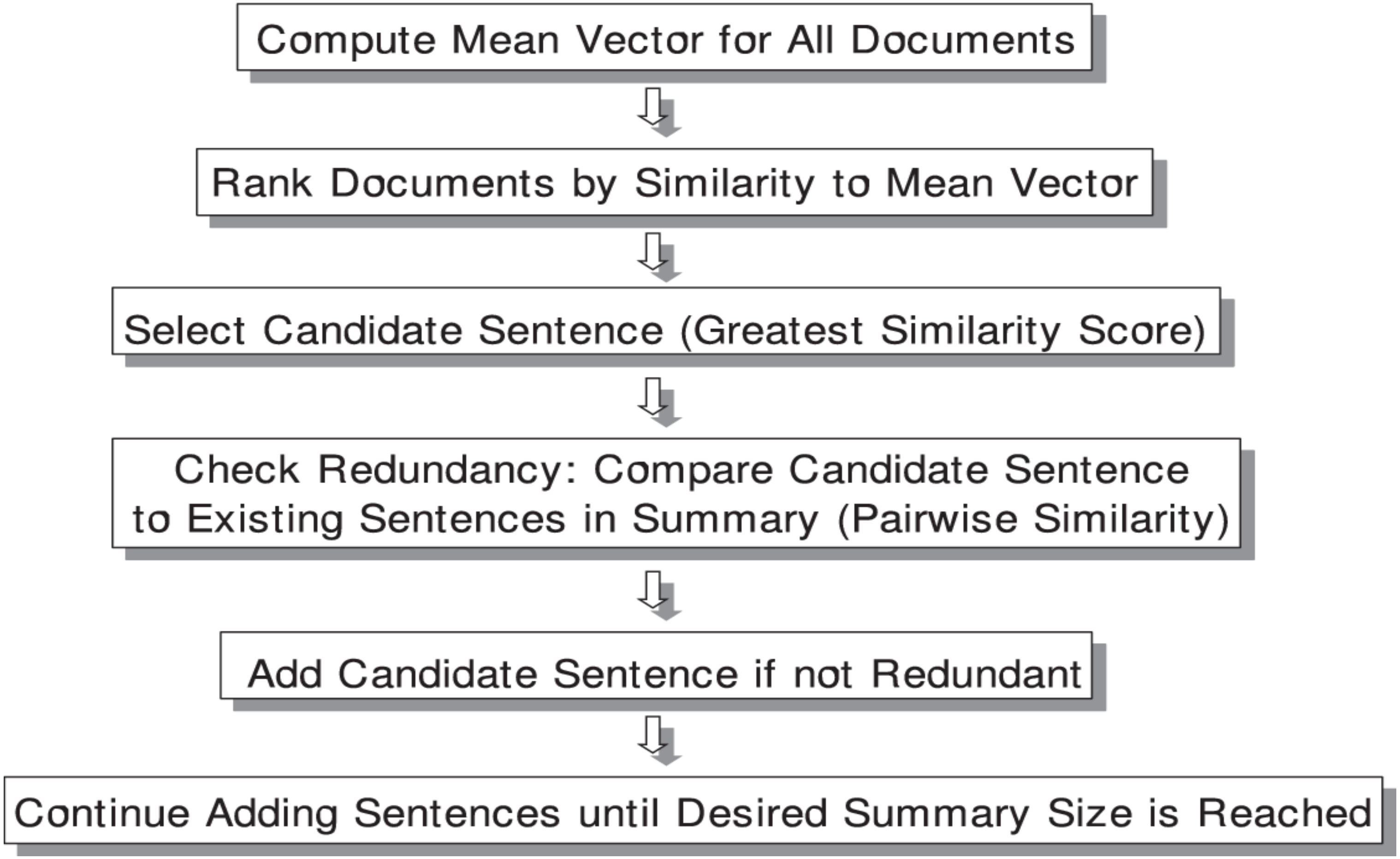
Check Redundancy: Compare Candidate Sentence to Existing Sentences in Summary (Pairwise Similarity)

Add Candidate Sentence if not Redundant

Continue Adding Sentences until Desired Summary Size is Reached

- Multiple documents also keep their order relative to their **time stamp**, with extracts from the earliest time stamp appearing first.

Text Summarization



Compute Mean Vector for All Documents

Rank Documents by Similarity to Mean Vector

Select Candidate Sentence (Greatest Similarity Score)

Check Redundancy: Compare Candidate Sentence to Existing Sentences in Summary (Pairwise Similarity)

Add Candidate Sentence if not Redundant

Continue Adding Sentences until Desired Summary Size is Reached

- Given a percentage threshold (e.g. 10%), sentences are added until the threshold is exceeded

Text Summarization

- Instead of a pure similarity score, a weighted score that includes other factors may be used, such as:
  - The **position** of the sentence in the document
    - Sentences appearing at the beginning of the document may be weighted higher than those in the middle. **(Luhn)**
  - The l**ength** of the sentence.

Hans Luhn, The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)

Text Summarization

- A related summarization method extracts sentences using **clustering**.
- The idea can be described as follows:
  - We partition sentences from a document (or multiple documents) into a number of clusters such that sentences within each cluster are similar.
  - Then we create the mean vector for each cluster.
  - Similar to the procedure in Fig. 9.3, one or more sentences can be selected that are closest to the mean vector from each cluster as its representative sentences.
  - The selected representative sentences from the clusters are displayed as the desired summarization.

Text Summarization

- In addition to the methods above, topic sentences can also be selected based on some linguistic heuristics:
  - A number of ideas have been proposed in the literature.
    - Titles, and sentences that occur in either very early or very late in a document and its paragraphs, tend to carry much more information about the main topics.

      → Select sentences simply based on their positions.
    - Another idea is to use linguistic cue phrases such as *in conclusion*, *the most important*, *or the purpose of the article*, which often indicate important topic sentences.

H. Edmundson. New methods in automatic extracting.
Journal of the ACM, 16(2):264–285, 1969.

Text Summarization

- Although linguistic heuristics such as those described above are helpful for sentence extraction, it is often difficult to determine the relative importance of sentences that are selected based on different heuristics.
- In order to facilitate a consistent ranking of sentences, an emerging trend in topic sentence extraction is to employ machine learning methods.
    - Trainable classifiers have been used to rank sentences based on features such as **cue phrase**, **location**, **sentence length**, **word frequency (tf-idf)** and **title**, etc.
        - Given a document, we may select a pool of top-ranked sentences based on outputs from the resulting classifier.
        - In order to obtain a more concise summary, an algorithm such as the one in Fig. 9.3 can be employed to eliminate redundant sentences from the pool.

A Closer Look at Extraction Methods

- Position in the text
  - lead method; optimal position policy
  - title/heading method
- Cue phrases in sentences
- Word frequencies throughout the text
- Cohesion: links among words
  - word co-occurrence
  - coreference
  - lexical chains
- Discourse structure of the text
- *Information Extraction*

Position-based Method

- **Claim**: Important sentences occur at the beginning (and/or end) of texts.
- **Lead method**: just take first sentence(s)!
- Experiments:
  - In 85% of 200 individual paragraphs the topic sentences occurred in initial position and in 7% in final position (Baxendale, 58).
  - Only 13% of the paragraphs of contemporary writers start with topic sentences (Donlan, 80).

Position-based Method

Individual contribution

- (Edmundson, 69)
  - **52**% recall & precision in combination with title (25% lead baseline)
- (Kupiec et al., 95)
  - 33% recall & precision
  - (24% lead baseline)
- (Teufel and Moens, 97)
  - 32% recall and precision (28% lead baseline)

Cumulative contribution

- (Edmundson, 69)
  - the best individual method
- (Kupiec et al., 95)
  - the best individual method
- (Teufel and Moens, 97)
  - increased performance by 10% when combined with the cue-based method

Optimum Position Policy (OPP)

- **Claim**: Important sentences are located at positions that are **genre-dependent**; these positions can be determined automatically through training (Lin and Hovy, 97).
  - **Corpus**: 13000 newspaper articles (ZIFF corpus).
  - **Step 1**: For each article, determine overlap between sentences and the index terms for the article.
  - **Step 2**: Determine a partial ordering over the locations where sentences containing important words occur: Optimal Position Policy (OPP)

Optimum Position Policy (OPP)

❖OPP for ZIFF corpus:

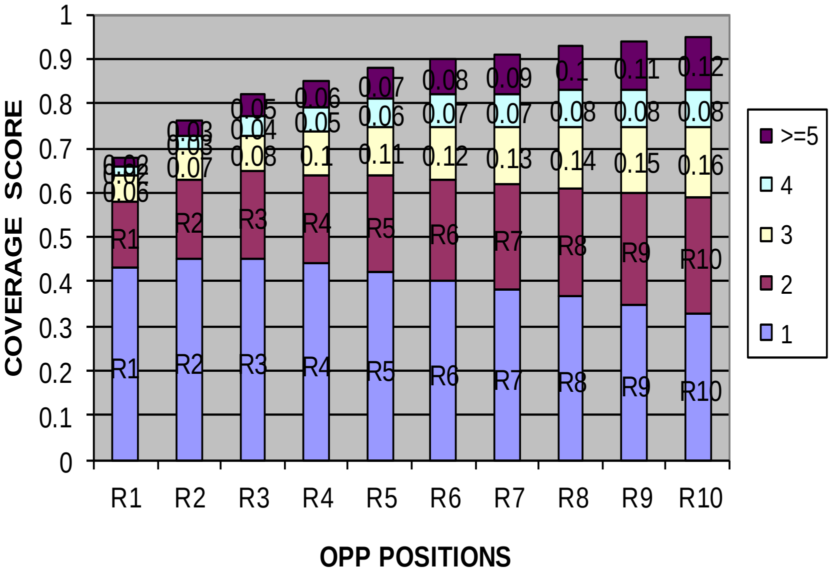
(T) > (P_2,S_1) > (P_3,S_1) > (P_2,S_2) > {(P_4,S_1),(P_5,S_1),(P_3,S_2)} >…

(T=title; P=paragraph; S=sentence)

❖OPP for *Wall Street Journal*: (T)>(P_1,S_1)>...

❖**Results**: testing corpus of 2900 articles: Recall=35%, Precision=38%.

❖**Results**: 10%-extracts cover 91% of the salient words.



Title-Based Method

- **Claim**: Words in titles and headings are positively relevant to summarization.
- Shown to be statistically valid at 99% level of significance (Edmundson, 69).
- Empirically shown to be useful in summarization systems.

Title-Based Method

Individual contribution

- (Edmundson, 69)
  - 40% recall & precision (25% lead baseline)
- (Teufel and Moens, 97)
  - 21.7% recall & precision (28% lead baseline)

Cumulative contribution

- (Edmundson, 69)
  - increased performance by 8% when combined with the cue-based methods.
- (Teufel and Moens, 97)
  - increased performance by 3% when combined with cue-, location-, position-, and word-frequency-based methods.

Cue-Phrase Method

- ❖ **Claim 1**: Important sentences contain ‘**bonus phrases**’ (i.e. positively relevant words/phrases), such as *significantly*, *In this paper we show*, and *In conclusion*, while non-important sentences contain ‘**stigma phrases**’ (i.e. negatively relevant phrase), such as *hardly* and *impossible*.
- ❖ **Claim 2**: These phrases can be detected automatically (Kupiec et al. 95; Teufel and Moens 97).
- ❖ **Method**: Add to sentence score if it contains a bonus word/phrase, penalize if it contains a stigma word/phrase.

H. Edmundson. New methods in automatic extracting.
Journal of the ACM, 16(2):264–285, 1969.

Cue-Phrase Method

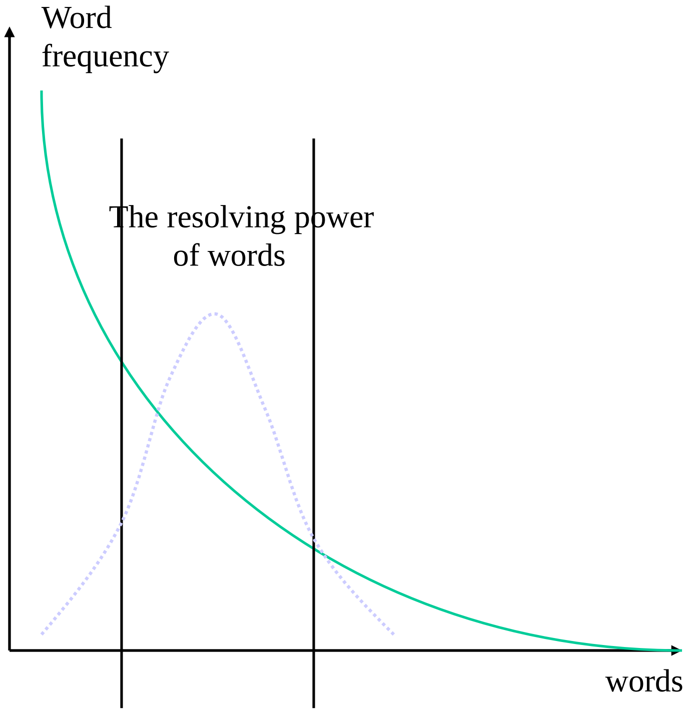
Individual contribution

- (Edmundson, 69)
  - 45% recall & precision (25% lead baseline)
- (Kupiec et al., 95)
  - 29% recall & precision (24% lead baseline)
- (Teufel and Moens, 97)
  - **55**% recall & precision (28% lead baseline)

Cumulative contribution

- (Edmundson, 69)
  - increased performance by 7% when combined with the title and position methods.
- (Kupiec et al., 95)
  - increased performance by 9% when combined with the position method.
- (Teufel and Moens, 97)
  - the best individual method.

Word-frequency-based method



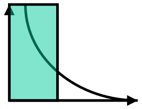
(Luhn, 59)

- ❖ **Claim**: Important sentences contain words that occur “somewhat” frequently.
- ❖ **Method**: Increase sentence score for each frequent word.
- ❖ **Evaluation**: Straightforward approach empirically shown to be mostly detrimental in summarization systems.

Word-frequency-based method

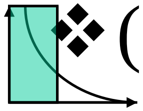
Individual contribution

- (Edmundson, 68)



  - 36% recall & precision (25% lead baseline)

- (Kupiec et al., 95)



  - 20% recall & precision (24% lead baseline)
- (Teufel and Moens, 97)

TF-IDF

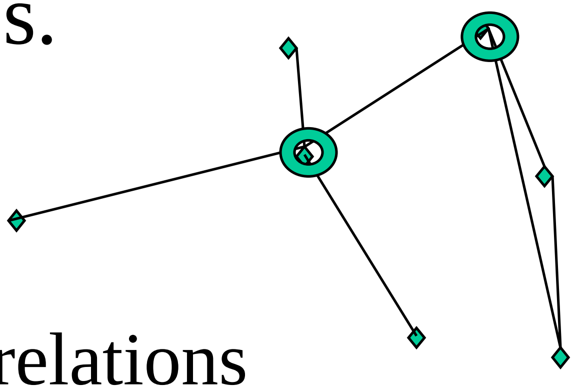
  - 17% recall & precision (28% lead baseline)

Cumulative contribution

- (Edmundson, 68)
  - decreased performance by 7% when combined with other methods
- (Kupiec et al., 95)
  - decreased performance by 2% when combined...
- (Teufel and Moens, 97)
  - increased performance by 0.2% when combined...

Cohesion-based Methods

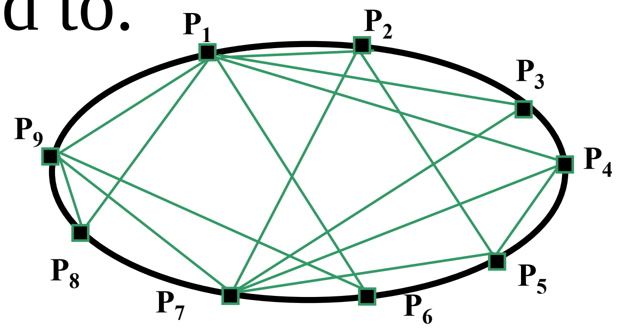
- **Claim**: Important sentences/paragraphs are the highest connected entities in more or less elaborate semantic structures.



- Classes of approaches
  - Word co-occurrences
  - Local salience/Grammatical relations
  - Co-reference
  - Lexical similarity (WordNet, lexical chains)
  - Combinations of the above

Cohesion: WORD Co-occurrence

- Apply IR methods at the document level: texts are collections of paragraphs (Salton et al., 94; Mitra et al., 97; Buckley and Cardie, 97):
  - Use a traditional, IR-based, word-based similarity measure to determine for each paragraph P_i the set S_i of paragraphs that P_i is related to.



- **Method**:
  - determine **relatedness** score S_i for each paragraph,
  - extract paragraphs with largest S_i scores.

Cohesion: WORD Co-occurrence

- **Study** (Mitra et al., 97):
  - Corpus: 50 articles from Funk and Wagner Encyclopedia.
  - Result: 46.0% overlap between two manual extracts.

	IR-based algorithm	Lead-based algorithm
Optimistic (best overlap)	45.6%	47.9%
Pessimistic (worst overlap)	30.7%	29.5%
Intersection	47.33%	50.0%
Union	55.16%	55.97%

Cohesion: Local Salience Method

- Assumes that important phrasal expressions are given by a combination of grammatical, syntactic, and contextual parameters (Boguraev and Kennedy, 97):

SUBJ:	80	iff the expression is a subject
EXST:	70	iff the expression is an existential construction
ACC:	50	iff the expression is a direct object
HEAD:	80	iff the expression is not contained in another phrase
ARG:	50	iff the expression is not contained in an adjunct

- No evaluation of the method.

Cohesion: Lexical Chains

Based on (Morris and Hirst, 91)

But Mr. Kenny’s move speeded up work on a **machine** which uses **micro-computers** to control the rate at which an *anaesthetic* is pumped into the blood of *patients* undergoing *surgery*. Such **machines** are nothing new. But Mr. Kenny’s **device** uses two **personal-computers** to achieve much closer monitoring of the **pump** feeding the *anaesthetic* into the *patient*. Extensive testing of the **equipment** has sufficiently impressed the authorities which regulate *medical* **equipment** in Britain, and, so far, four other countries, to make this the first such **machine** to be licensed for commercial sale to *hospitals*.

Cohesion: Lexical Chains

- Assumes that important sentences are those that are ‘traversed’ by **strong** chains (Barzilay and Elhadad, 97).

  Strength(C) = length(C) - #DistinctOccurrences(C)

  - For each chain, choose the first sentence that is traversed by the chain and that uses a representative set of concepts from that chain.

[Jing et al., 98] corpus	LC algorithm		Lead-based algorithm	
	Recall	Prec	Recall	Prec
10% cutoff	67%	61%	82.9%	63.4%
20% cutoff	64%	47%	70.9%	46.9%

Cohesion: Coreference Method

- Build co-reference chains (noun/event identity, part-whole relations) between
  - *query and document* - In the context of query-based summarization
  - title and document
  - sentences within document
- Important sentences are those traversed by a large number of chains:
  - a preference is imposed on chains (*query* > title > doc)
- Evaluation: 67% F-score for relevance (SUMMAC, 98). (Baldwin and Morton, 98)

Cohesion: Connectedness Method

- Map texts into graphs:
  - The nodes of the graph are the words of the text.
  - Arcs represent adjacency, grammatical, co-reference, and lexical similarity-based relations.
- Associate importance scores to words (and sentences) by applying *tf-idf*.
- Assume that important words/sentences are those with the highest scores.

[Marcu,97] corpus	**TF-IDF method**	**Spreading activation**
10% cutoff F-score	25.2%	32.4%
20% cutoff F-score	35.8%	45.4%

(Mani and Bloedorn, 97)

Discourse-Based Method

- **Claim**: The multi-sentence coherence structure of a text can be constructed, and the ‘centrality’ of the textual units in this structure reflects their importance.
- Tree-like representation of texts in the style of **Rhetorical Structure Theory** (Mann and Thompson,88).
- Use the discourse representation in order to determine the most important textual units. Attempts:
  - (Ono et al., 94) for Japanese.
  - (Marcu, 97) for English.

Rhetorical Parsing: An Example

[*With* its distant orbit {– 50 percent farther from the sun than Earth –} and slim atmospheric blanket,^{1}] [Mars experiences frigid weather conditions.^{2}] [Surface temperatures typically average about –60 degrees Celsius (–76 degrees Fahrenheit) at the equator and can dip to –123 degrees C near the poles.^{3}] [Only the midday sun at tropical latitudes is warm enough to thaw ice on occasion,^{4}] [*but* any liquid water formed that way would evaporate almost instantly^{5}] [*because* of the low atmospheric pressure.^{6}]

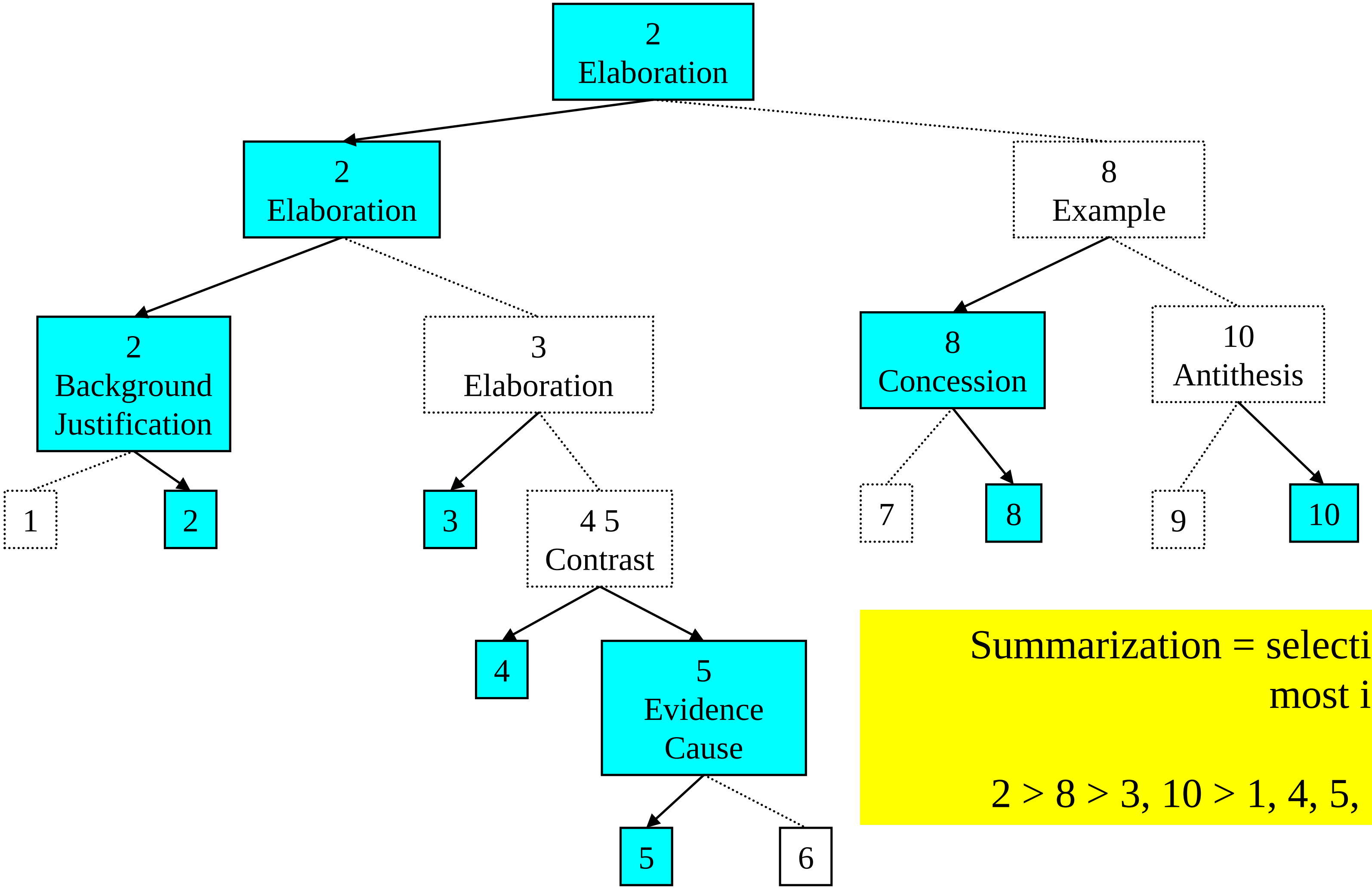
Rhetorical Parsing: An Example

[*Although* the atmosphere holds a small amount of water, and water-ice clouds sometimes develop,^{7}] [most Martian weather involves blowing dust or carbon dioxide.^{8}] [Each winter, *for example*, a blizzard of frozen carbon dioxide rages over one pole, and a few meters of this dry-ice snow accumulate as previously frozen carbon dioxide evaporates from the opposite polar cap.^{9}] [*Yet* even on the summer pole, {*where* the sun remains in the sky all day long,} temperatures never warm enough to melt frozen water.^{10}]

Rhetorical Parsing: An Example

- Use discourse markers to hypothesize rhetorical relations
  - rhet_rel(CONTRAST, 4, 5) ⊕ rhet_rel(CONTRAST, 4, 6)
  - rhet_rel(EXAMPLE, 9, [7,8]) ⊕ rhet_rel(EXAMPLE, 10, [7,8])
- Use semantic similarity to hypothesize rhetorical relations
  - if similar(u_1,u_2) then
    rhet_rel(ELABORATION, u_2, u_1) ⊕ rhet_rel(BACKGROUND, u_1,u_2)
    else
    rhet_rel(JOIN, u_1, u_2)
- Use the hypotheses in order to derive a valid discourse representation of the original text.

Rhetorical Parsing: An Example



Summarization = selection of the most important units

2 > 8 > 3, 10 > 1, 4, 5, 7, 9 > 6

Discourse Method: Evaluation

(using a combination of heuristics for rhetorical parsing disambiguation)

Reduction	Method	Recall	Precision	F-score
10%	Humans	83.20%	75.95%	79.41%
	Program	68.33%	84.16%	75.42%
	Lead	82.91%	63.45%	71.89%
20%	Humans	82.83%	64.93%	72.80%
	Program	59.51%	72.11%	65.21%
	Lead	70.91%	46.96%	56.50%

TREC
Corpus

Level	Method	Recall	Precision	F-score
Clause	Humans	72.66%	69.63%	71.27%
	Program	67.57%	73.53%	70.42%
	Lead	39.68%	39.68%	39.68%
Sentence	Humans	78.11%	79.37%	78.73%
	Program	69.23%	64.29%	66.67%
	Lead	54.22%	54.22%	54.22%

Scientific American
Corpus

Centrality Analysis in Summarization

- Motivation: capture the most **central** words in a document or cluster.
  - 2 popular methods:
    - TextRank
    - LexRank

TextRank & LexRank

- The TextRank algorithm exploits the structure of the text itself to determine keyphrases that appear "central" to the text in the same way that **PageRank** selects important Web pages.
  - Recall this is related to centrality analysis (eigenvector centrality) in social networks.

- LexRank is an algorithm essentially identical to TextRank, and both can be used for document summarization.
  - The two methods were developed by different groups at the same time, and LexRank simply focused on summarization, but could just as easily be used for **keyphrase** extraction or any other NLP ranking task.

R. Mihalcea and P. Tarau. Textrank: Bringing order into texts.
In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 404–411, 2004.

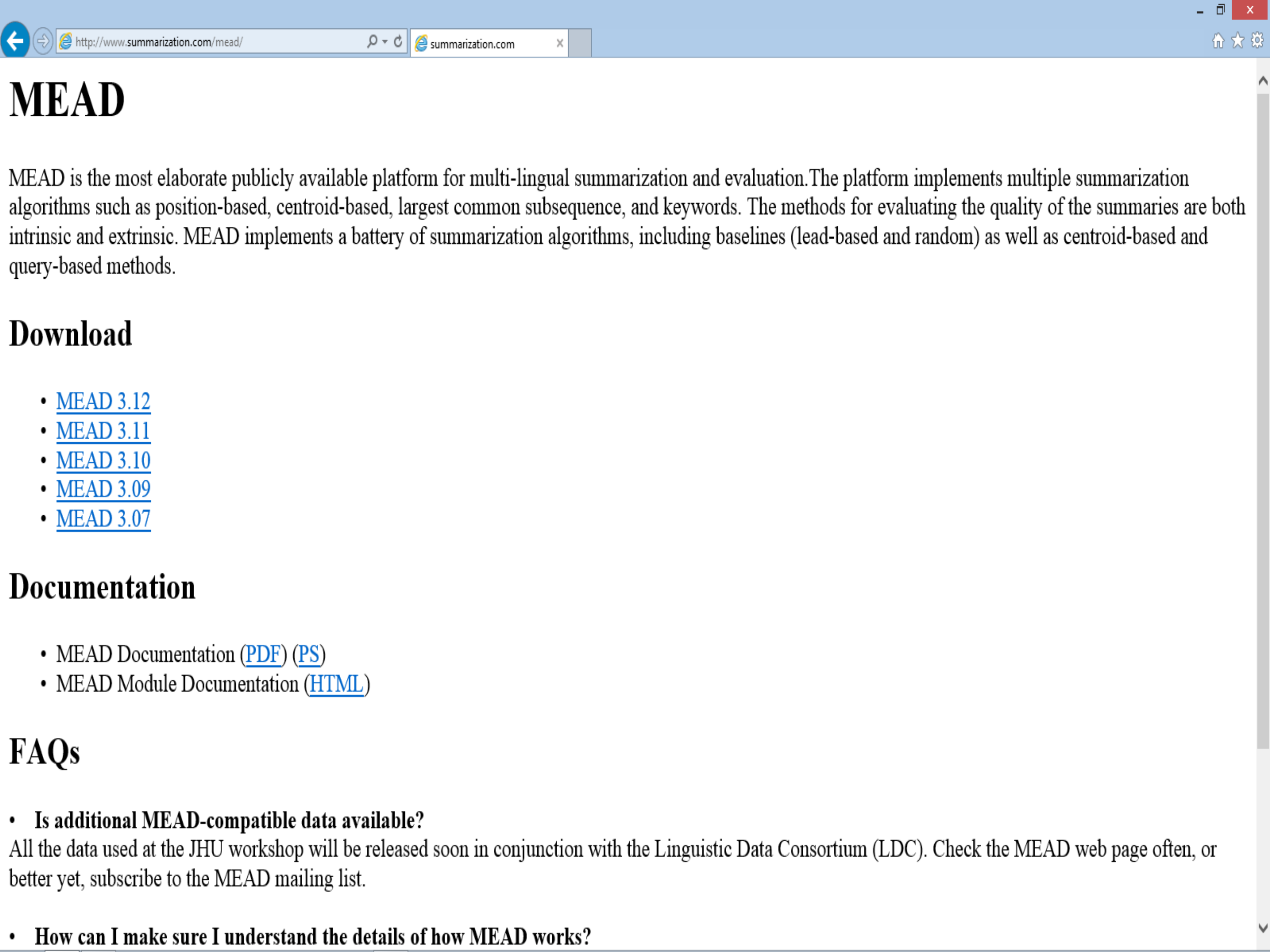
TextRank & LexRank

- In both LexRank and TextRank, a graph is constructed by creating a **vertex** for each sentence in the document. The **edges** between sentences are based on some form of semantic similarity or content overlap.
  - While LexRank uses cosine similarity of TF-IDF vectors, TextRank uses a very similar measure based on the number of words two sentences have in common (normalized by the sentences' lengths). → Jaccard
  - A summary is formed by combining the top ranking sentences, using a threshold or length cutoff to limit the size of the summary.

G. Erkan and D. Radev. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 2004.

MEAD

- It is worth noting that TextRank was applied to summarization exactly as described here, while LexRank was used as part of a larger summarization system (**MEAD**).
- MEAD combines the LexRank score with other features like sentence position and length using a linear combination with either user-specified or automatically tuned weights. (See next slide)



MEAD

MEAD is the most elaborate publicly available platform for multi-lingual summarization and evaluation.The platform implements multiple summarization algorithms such as position-based, centroid-based, largest common subsequence, and keywords. The methods for evaluating the quality of the summaries are both intrinsic and extrinsic. MEAD implements a battery of summarization algorithms, including baselines (lead-based and random) as well as centroid-based and query-based methods.

Download

- MEAD 3.12
- MEAD 3.11
- MEAD 3.10
- MEAD 3.09
- MEAD 3.07

Documentation

- MEAD Documentation (PDF) (PS)
- MEAD Module Documentation (HTML)

FAQs

- **Is additional MEAD-compatible data available?**

All the data used at the JHU workshop will be released soon in conjunction with the Linguistic Data Consortium (LDC). Check the MEAD web page often, or better yet, subscribe to the MEAD mailing list.

- **How can I make sure I understand the details of how MEAD works?**

3. Features

The following features are provided with MEAD. They are all computed on a sentence-by-sentence basis.

- Centroid: cosine overlap with the centroid vector of the cluster (Radev et al., 2004),
- SimWithFirst: cosine overlap with the first sentence in the document (or with the title, if it exists),
- Length: 1 if the length of the sentence is above a given threshold and 0 otherwise,
- RealLength: the length of the sentence in words,
- Position: the position of the sentence in the document,
- QueryOverlap: cosine overlap with a query sentence or phrase,
- KeyWordMatch: full match from a list of keywords,
- LexPageRank: eigenvector centrality of the sentence on the lexical connectivity matrix with a defined threshold.

LexRank

- Another important distinction is that TextRank was used for single document summarization, while LexRank has been applied to multi-document summarization.
  - The task remains the same in both cases, only the number of sentences to choose from has grown.
  - However, when summarizing multiple documents, there is a greater risk of selecting duplicate or highly redundant sentences to place in the same summary.

LexRank: An Example

1 (d1s1) Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.

2 (d2s1) Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq rejects cooperating with the United Nations except on the issue of lifting the blockade imposed upon it since the year 1990.

3 (d2s2) Ramadan told reporters in Baghdad that "Iraq cannot deal positively with whoever represents the Security Council unless there was a clear stance on the issue of lifting the blockade off of it.

4 (d2s3) Baghdad had decided late last October to completely cease cooperating with the inspectors of the United Nations Special Commission (UNSCOM), in charge of disarming Iraq's weapons, and whose work became very limited since the fifth of August, and announced it will not resume its cooperation with the Commission even if it were subjected to a military operation.

5 (d3s1) The Russian Foreign Minister, Igor Ivanov, warned today, Wednesday against using force against Iraq, which will destroy, according to him, seven years of difficult diplomatic work and will complicate the regional situation in the area.

6 (d3s2) Ivanov contended that carrying out air strikes against Iraq, who refuses to cooperate with the United Nations inspectors, ``will end the tremendous work achieved by the international group during the past seven years and will complicate the situation in the region."

7 (d3s3) Nevertheless, Ivanov stressed that Baghdad must resume working with the Special Commission in charge of disarming the Iraqi weapons of mass destruction (UNSCOM).

8 (d4s1) The Special Representative of the United Nations Secretary-General in Baghdad, Prakash Shah, announced today, Wednesday, after meeting with the Iraqi Deputy Prime Minister Tariq Aziz, that Iraq refuses to back down from its decision to cut off cooperation with the disarmament inspectors.

9 (d5s1) British Prime Minister Tony Blair said today, Sunday, that the crisis between the international community and Iraq ``did not end" and that Britain is still ``ready, prepared, and able to strike Iraq."

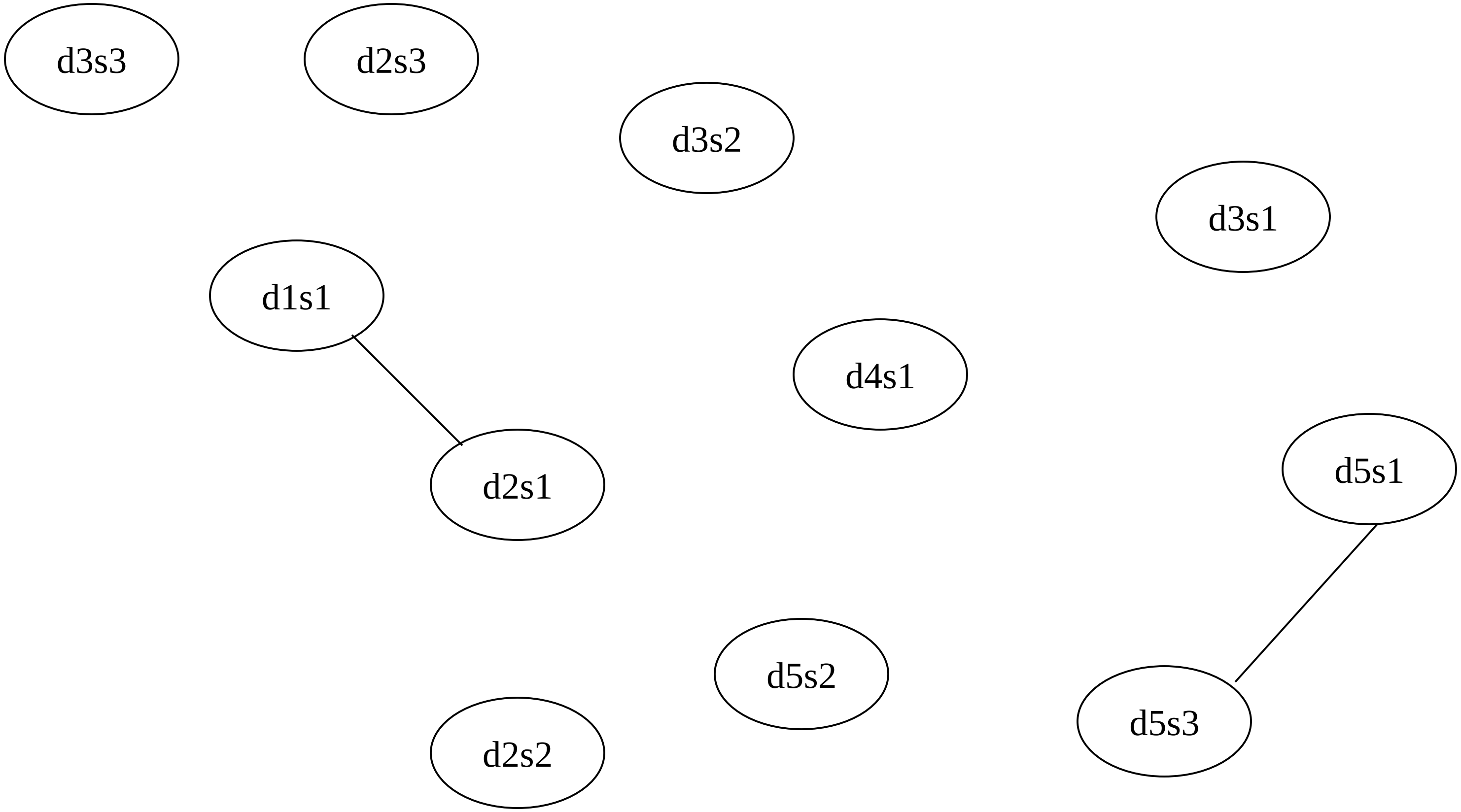
10 (d5s2) In a gathering with the press held at the Prime Minister's office, Blair contended that the crisis with Iraq ``will not end until Iraq has absolutely and unconditionally respected its commitments" towards the United Nations.

11 (d5s3) A spokesman for Tony Blair had indicated that the British Prime Minister gave permission to British Air Force Tornado planes stationed in Kuwait to join the aerial bombardment against Iraq.

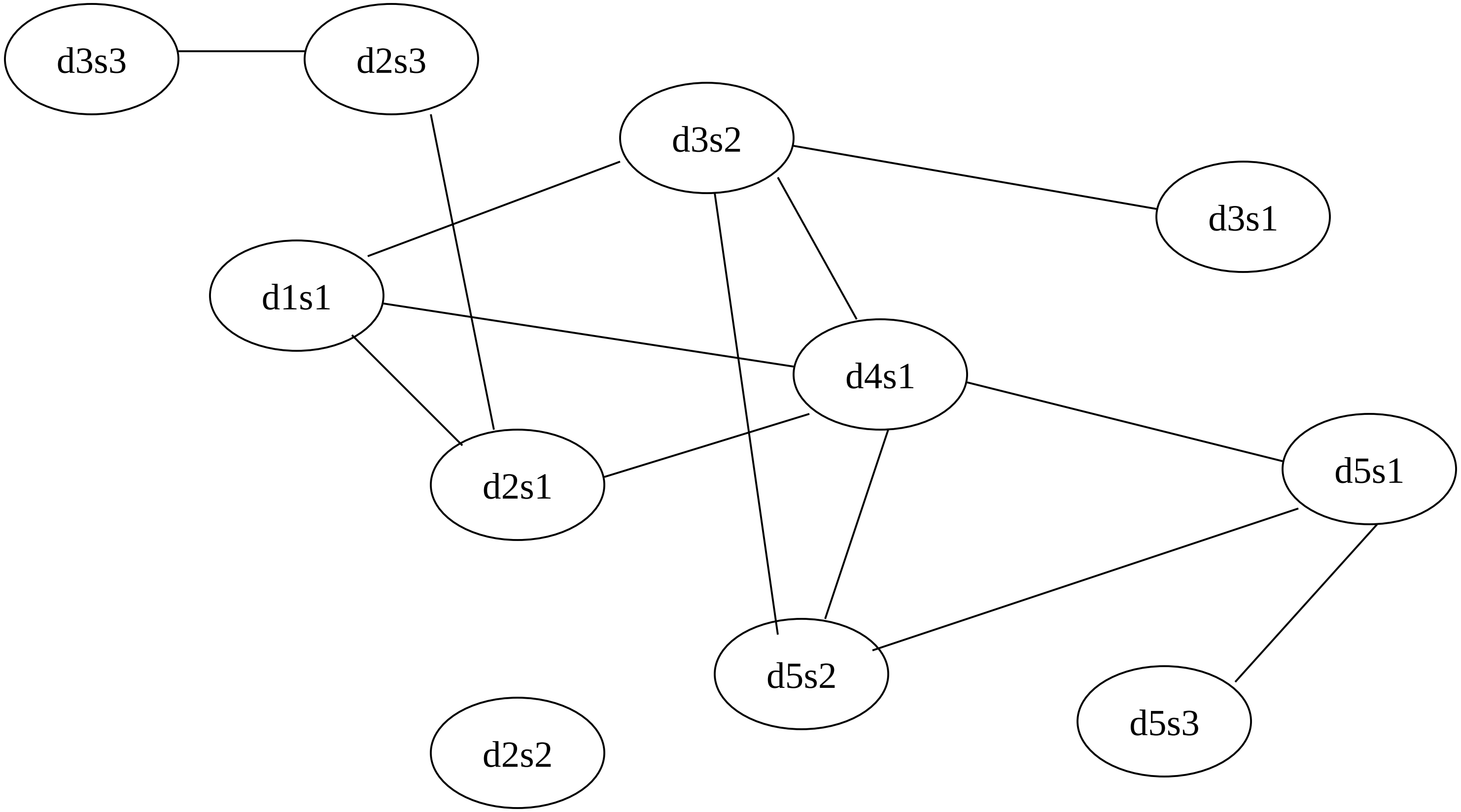
Cosine Centrality

	1	2	3	4	5	6	7	8	9	10	11
1	1.00	0.45	0.02	0.17	0.03	0.22	0.03	0.28	0.06	0.06	0.00
2	0.45	1.00	0.16	0.27	0.03	0.19	0.03	0.21	0.03	0.15	0.00
3	0.02	0.16	1.00	0.03	0.00	0.01	0.03	0.04	0.00	0.01	0.00
4	0.17	0.27	0.03	1.00	0.01	0.16	0.28	0.17	0.00	0.09	0.01
5	0.03	0.03	0.00	0.01	1.00	0.29	0.05	0.15	0.20	0.04	0.18
6	0.22	0.19	0.01	0.16	0.29	1.00	0.05	0.29	0.04	0.20	0.03
7	0.03	0.03	0.03	0.28	0.05	0.05	1.00	0.06	0.00	0.00	0.01
8	0.28	0.21	0.04	0.17	0.15	0.29	0.06	1.00	0.25	0.20	0.17
9	0.06	0.03	0.00	0.00	0.20	0.04	0.00	0.25	1.00	0.26	0.38
10	0.06	0.15	0.01	0.09	0.04	0.20	0.00	0.20	0.26	1.00	0.12
11	0.00	0.00	0.00	0.01	0.18	0.03	0.01	0.17	0.38	0.12	1.00

Cosine Centrality (t=0.3)

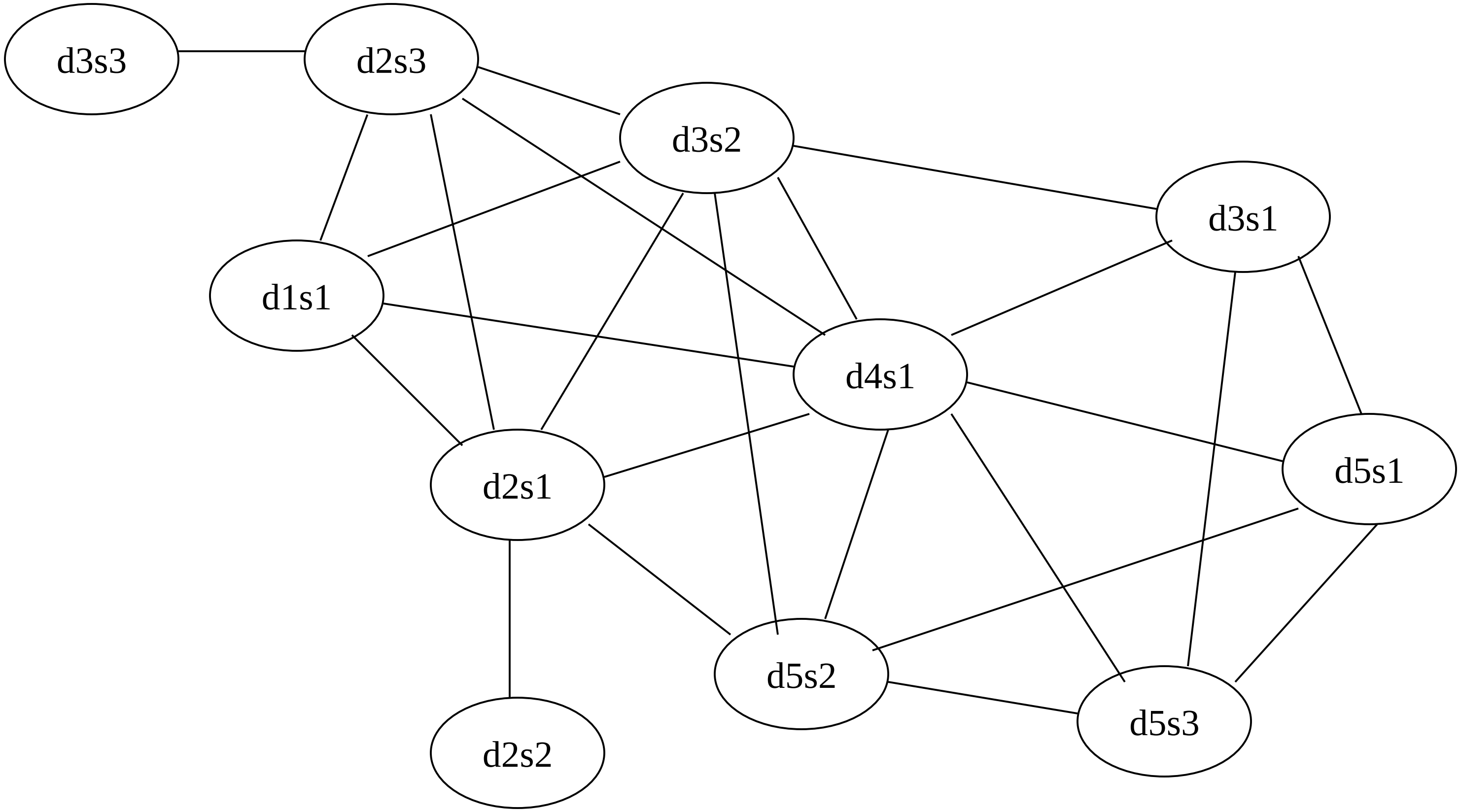


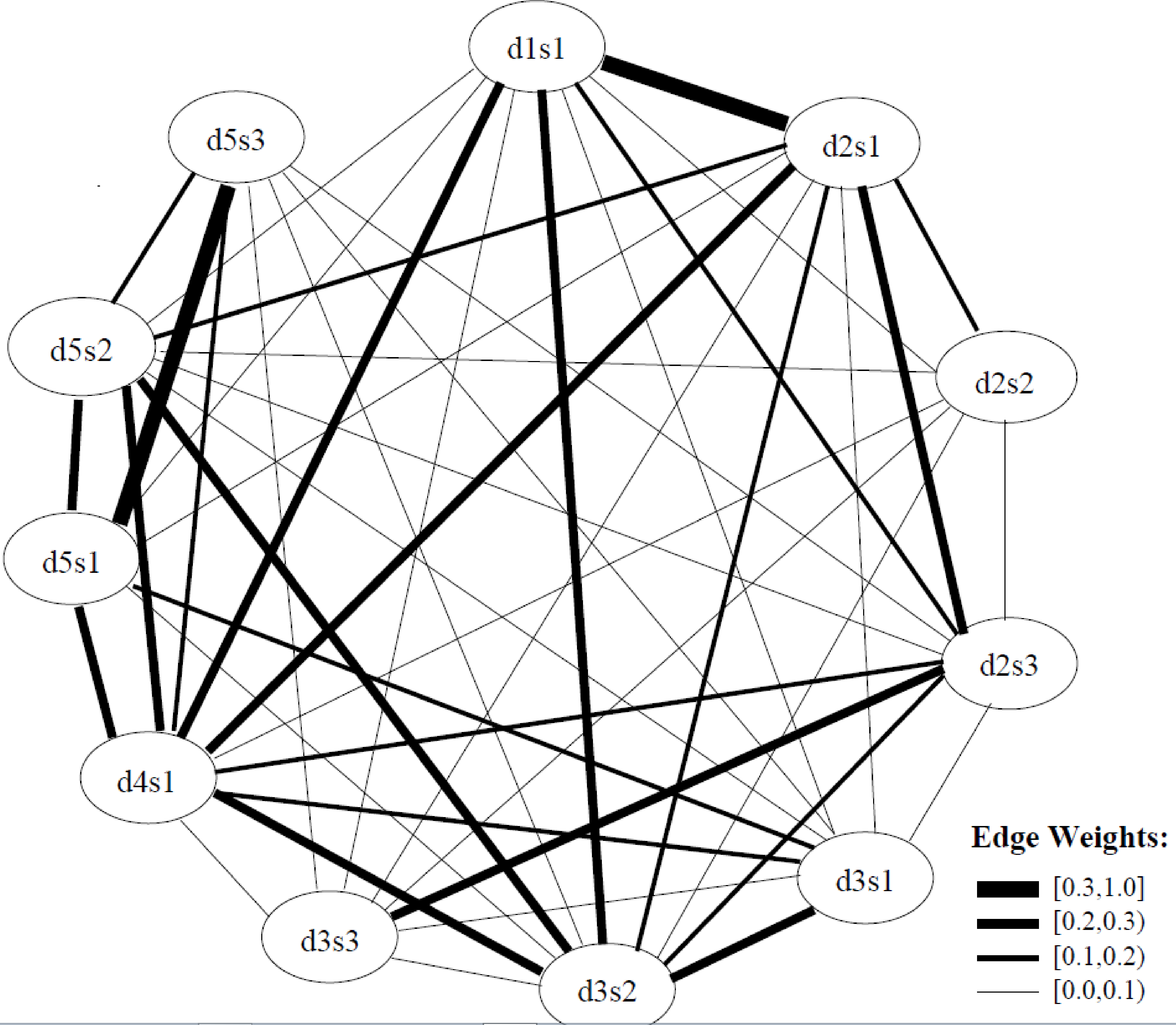
Cosine Centrality (t=0.2)



Cosine Centrality (t=0.1)

Sentences vote for the most central sentence!




d1s1
d5s3
d2s1
d5s2
d2s2
d5s1
d2s3
d4s1
d3s1
d3s3
d3s2
Edge Weights:
[0.3,1.0]
[0.2,0.3)
[0.1,0.2)
[0.0,0.1)

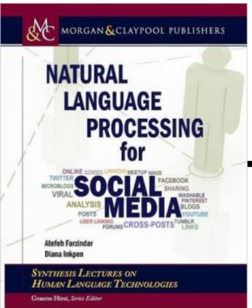
Cosine Centrality vs. Centroid

ID	LPR (0.1)	LPR (0.2)	LPR (0.3)	Centroid
d1s1	0.6007	0.6944	0.0909	0.7209
d2s1	0.8466	0.7317	0.0909	0.7249
d2s2	0.3491	0.6773	0.0909	0.1356
d2s3	0.7520	0.6550	0.0909	0.5694
d3s1	0.5907	0.4344	0.0909	0.6331
d3s2	0.7993	0.8718	0.0909	0.7972
d3s3	0.3548	0.4993	0.0909	0.3328
d4s1	1.0000	1.0000	0.0909	0.9414
d5s1	0.5921	0.7399	0.0909	0.9580
d5s2	0.6910	0.6967	0.0909	1.0000
d5s3	0.5921	0.4501	0.0909	0.7902

Text Summarization & Social Media

- ❖ Automatic summarization from multiple social media sources is a highly active research topic that aims at reducing and aggregating the amount of information presented to users.
- ❖ Due to the scale of social media, combined with the high level of noise present in social media, most texts are certain to be irrelevant for any particular information need. (See next slide)
  - ❖ Hence, some forms of information retrieval and/or filtering are generally a prerequisite to summarization.
  - ❖ Also, there is less of a focus on what individual "documents" are about, but rather how they can contribute to a summary of some real-world phenomenon.

    → **Multi-document summarization**

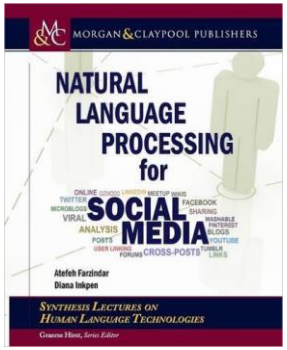

MORGAN & CLAYPOOL PUBLISHERS
NATURAL LANGUAGE PROCESSING for SOCIAL MEDIA
Atefeh Farzindar
Diana Inkpen
SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES
Graeme Hirst, Series Editor

Other Forms of Noise (Revisited)

- Social media are also much noisier than traditional print media.
  - Like much else on the Internet, social networks are plagued with **spam**, **ads**, and all manner of other unsolicited, irrelevant, or distracting content.
  - Even by ignoring these forms of noise, much of the genuine, legitimate content on social media can be seen as irrelevant with respect to most information needs.
- In a study, the authors collected over 40,000 tweets. Only 36% of the tweets were rated as “worth reading.”
  - The least valued tweets were so-called presence maintenance posts (e.g., “Hullo twitter!”).
  - Pre-processing to filter out spam and other irrelevant content, or models that are better capable of coping with noise, are essential in mining social media texts.

Text Summarization & Social Media

- In order to summarize multiple Twitter messages on the same topic, **multi-document summarization**, was attempted by several researchers, for example by adapting tools used in multi-document summarization for newspaper articles or other kinds of texts [Inouye and Kalita, 2011, Sharifi et al., 2010].
  - Solutions may involve clustering the important sentences picked out from the various messages and using only a few representative sentences from each cluster.
  - Machine learning techniques can be used to learn how to rank the selected tweets [Duan et al., 2010].



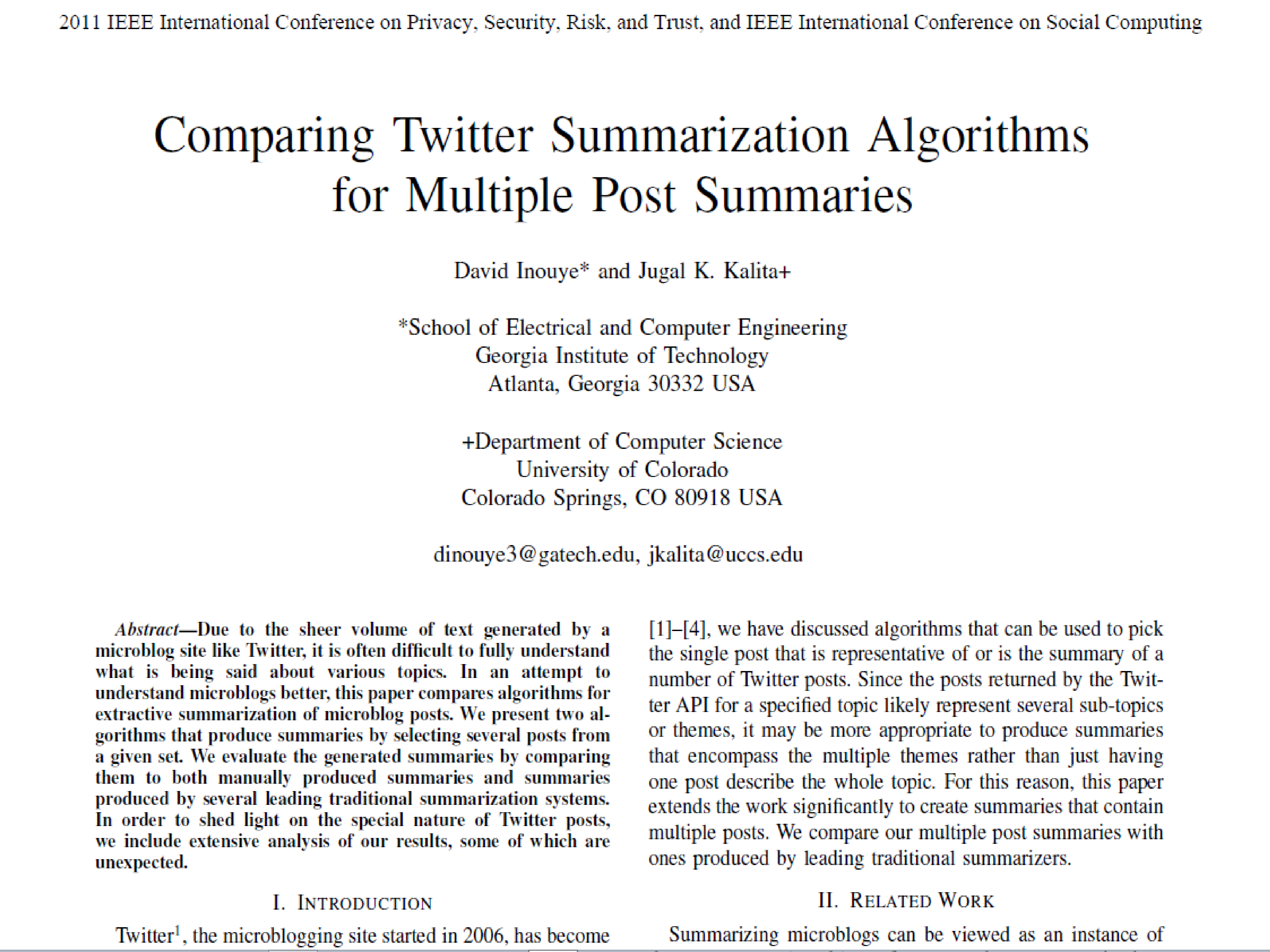
M&C MORGAN & CLAYPOOL PUBLISHERS

NATURAL LANGUAGE PROCESSING for SOCIAL MEDIA

Atefeh Farzindar
Diana Inkpen

SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES

Graeme Hirst, *Series Editor*



Comparing Twitter Summarization Algorithms for Multiple Post Summaries

David Inouye* and Jugal K. Kalita+

*School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332 USA

+Department of Computer Science
University of Colorado
Colorado Springs, CO 80918 USA

dinouye3@gatech.edu, jkalita@uccs.edu

***Abstract*—Due to the sheer volume of text generated by a microblog site like Twitter, it is often difficult to fully understand what is being said about various topics. In an attempt to understand microblogs better, this paper compares algorithms for extractive summarization of microblog posts. We present two algorithms that produce summaries by selecting several posts from a given set. We evaluate the generated summaries by comparing them to both manually produced summaries and summaries produced by several leading traditional summarization systems. In order to shed light on the special nature of Twitter posts, we include extensive analysis of our results, some of which are unexpected.**

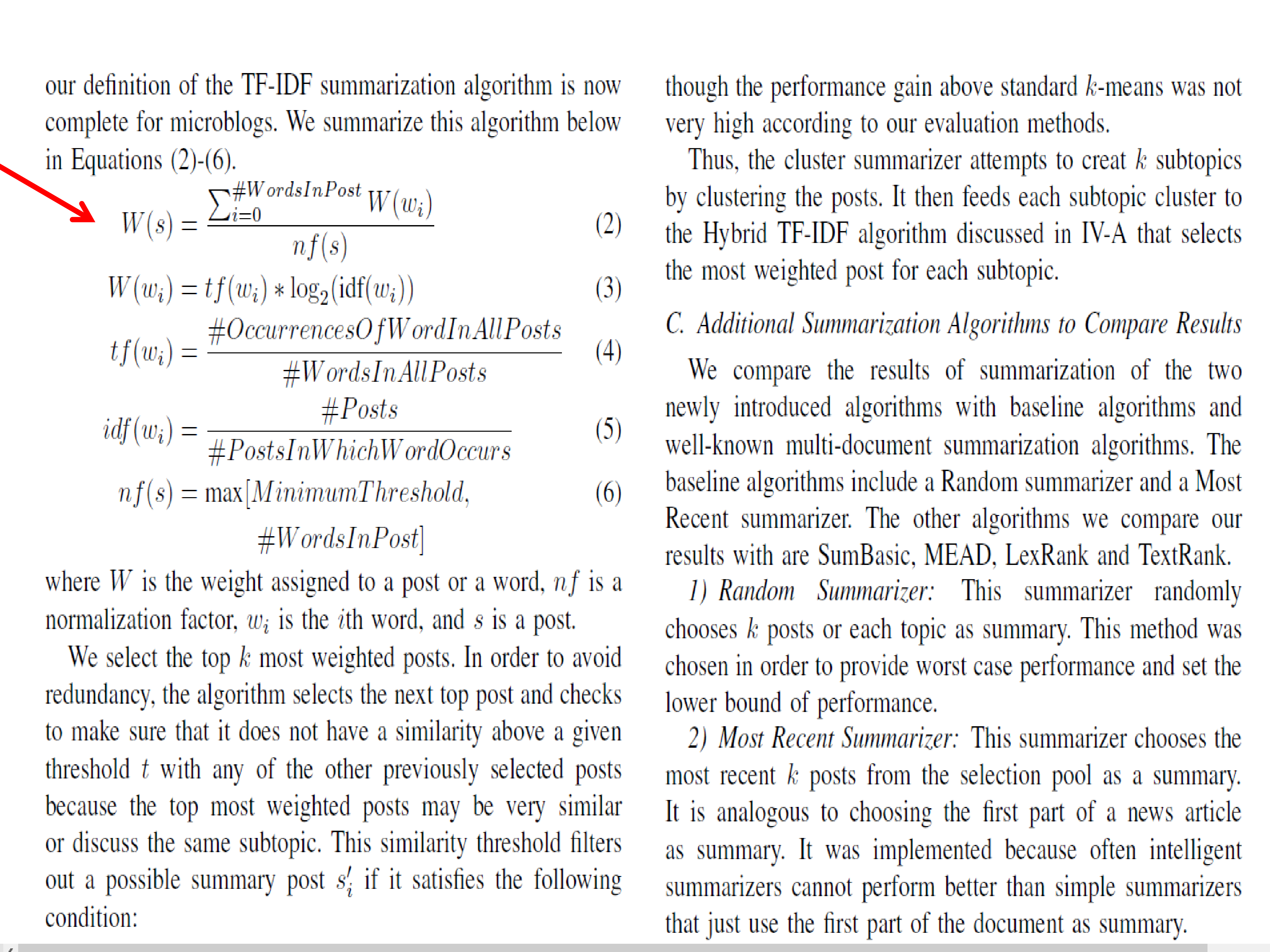
I. Introduction

Twitter[1], the microblogging site started in 2006, has become

[1]–[4], we have discussed algorithms that can be used to pick the single post that is representative of or is the summary of a number of Twitter posts. Since the posts returned by the Twitter API for a specified topic likely represent several sub-topics or themes, it may be more appropriate to produce summaries that encompass the multiple themes rather than just having one post describe the whole topic. For this reason, this paper extends the work significantly to create summaries that contain multiple posts. We compare our multiple post summaries with ones produced by leading traditional summarizers.

II. Related Work

Summarizing microblogs can be viewed as an instance of



our definition of the TF-IDF summarization algorithm is now complete for microblogs. We summarize this algorithm below in Equations (2)-(6).

$$W(s) = \frac{\sum_{i=0}^{\#WordsInPost} W(w_i)}{nf(s)} \tag{2}$$

$$W(w_i) = tf(w_i) * \log_2(\mathrm{idf}(w_i)) \tag{3}$$

$$tf(w_i) = \frac{\#OccurrencesOfWordInAllPosts}{\#WordsInAllPosts} \tag{4}$$

$$idf(w_i) = \frac{\#Posts}{\#PostsInWhichWordOccurs} \tag{5}$$

$$nf(s) = \max[MinimumThreshold, \#WordsInPost] \tag{6}$$

where W is the weight assigned to a post or a word, nf is a normalization factor, w_i is the ith word, and s is a post.

We select the top k most weighted posts. In order to avoid redundancy, the algorithm selects the next top post and checks to make sure that it does not have a similarity above a given threshold t with any of the other previously selected posts because the top most weighted posts may be very similar or discuss the same subtopic. This similarity threshold filters out a possible summary post s_i' if it satisfies the following condition:

though the performance gain above standard k-means was not very high according to our evaluation methods.

Thus, the cluster summarizer attempts to creat k subtopics by clustering the posts. It then feeds each subtopic cluster to the Hybrid TF-IDF algorithm discussed in IV-A that selects the most weighted post for each subtopic.

C. Additional Summarization Algorithms to Compare Results

We compare the results of summarization of the two newly introduced algorithms with baseline algorithms and well-known multi-document summarization algorithms. The baseline algorithms include a Random summarizer and a Most Recent summarizer. The other algorithms we compare our results with are SumBasic, MEAD, LexRank and TextRank.

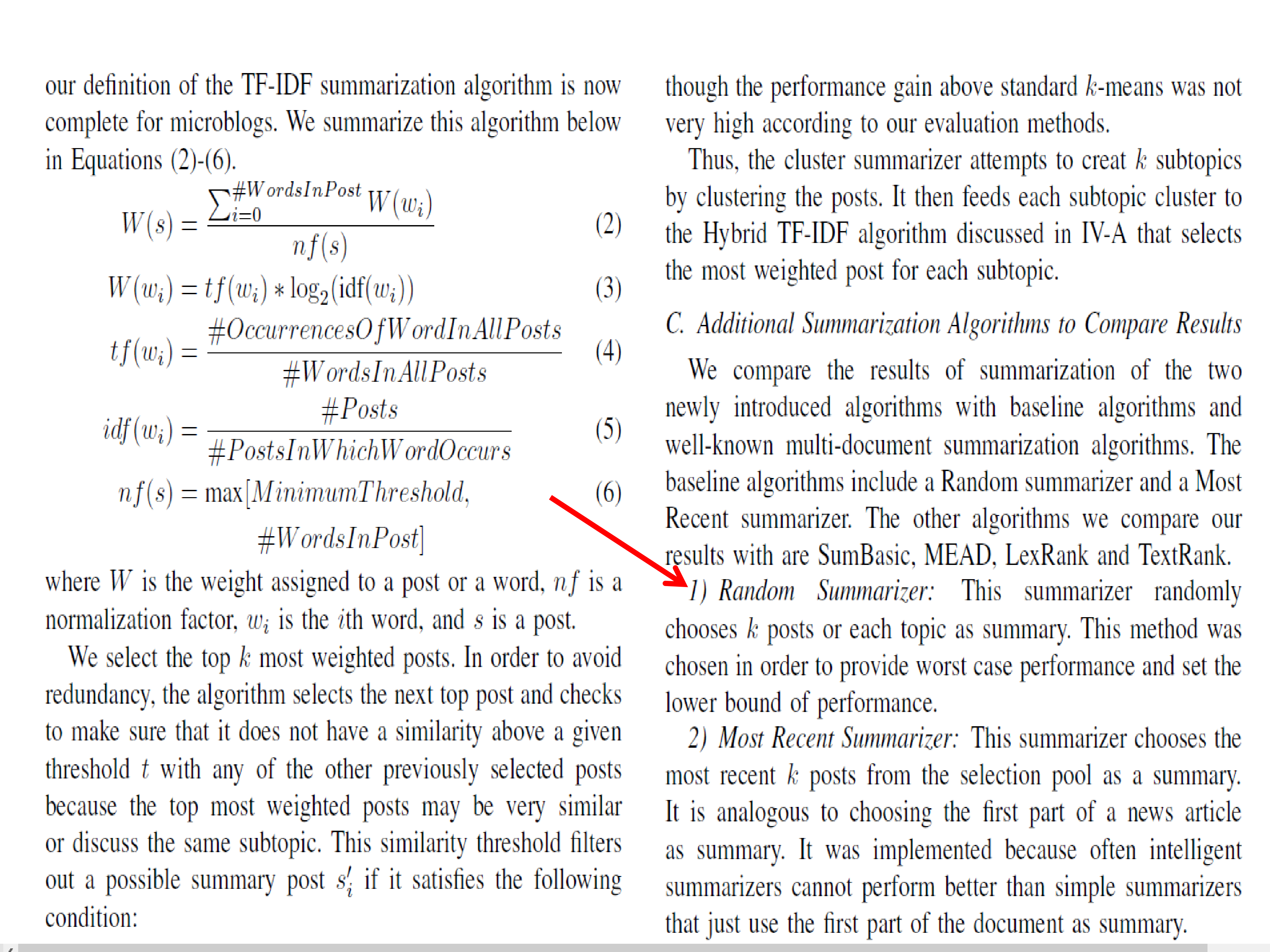
1) Random Summarizer: This summarizer randomly chooses k posts or each topic as summary. This method was chosen in order to provide worst case performance and set the lower bound of performance.

2) Most Recent Summarizer: This summarizer chooses the most recent k posts from the selection pool as a summary. It is analogous to choosing the first part of a news article as summary. It was implemented because often intelligent summarizers cannot perform better than simple summarizers that just use the first part of the document as summary.

B. Cluster Summarizer

We develop another method for summarizing a set of Twitter posts. Similar to [15] and [16], we first cluster the tweets into k clusters based on a similarity measure and then summarize each cluster by picking the most weighted post as determined by the Hybrid TF-IDF weighting described in Section IV-A.

During preliminary tests, we evaluated how well different clustering algorithms would work on Twitter posts using the weights computed by the Hybrid TF-IDF algorithm and the cosine similarity measure. We implemented two variations of the k-means algorithm: bisecting k-means [18] and k-means++ [19]. The bisecting k-means algorithm initially divides the input into two clusters and then divides the largest cluster into two smaller clusters. This splitting is repeated until the kth cluster is formed. The k-means++ algorithm is similar to the regular k-means algorithm except that it chooses the initial centroids differently. It picks an initial centroid c_1 from the set of vertices V randomly. It then chooses the next centroid c_i, selecting $c_i = v' \in V$ with the probability $\frac{D(v')^2}{\sum_{v \in V} D(v)^2}$ where $D(v)$ is the shortest Euclidean distance from v to the closest center which is already known. It repeats this selection process until k initial centroids have been chosen. After trying these



our definition of the TF-IDF summarization algorithm is now complete for microblogs. We summarize this algorithm below in Equations (2)-(6).

$$W(s) = \frac{\sum_{i=0}^{\#WordsInPost} W(w_i)}{nf(s)} \tag{2}$$

$$W(w_i) = tf(w_i) * \log_2(\mathrm{idf}(w_i)) \tag{3}$$

$$tf(w_i) = \frac{\#OccurrencesOfWordInAllPosts}{\#WordsInAllPosts} \tag{4}$$

$$idf(w_i) = \frac{\#Posts}{\#PostsInWhichWordOccurs} \tag{5}$$

$$nf(s) = \max[MinimumThreshold, \#WordsInPost] \tag{6}$$

where W is the weight assigned to a post or a word, nf is a normalization factor, w_i is the ith word, and s is a post.

We select the top k most weighted posts. In order to avoid redundancy, the algorithm selects the next top post and checks to make sure that it does not have a similarity above a given threshold t with any of the other previously selected posts because the top most weighted posts may be very similar or discuss the same subtopic. This similarity threshold filters out a possible summary post s_i' if it satisfies the following condition:

though the performance gain above standard k-means was not very high according to our evaluation methods.

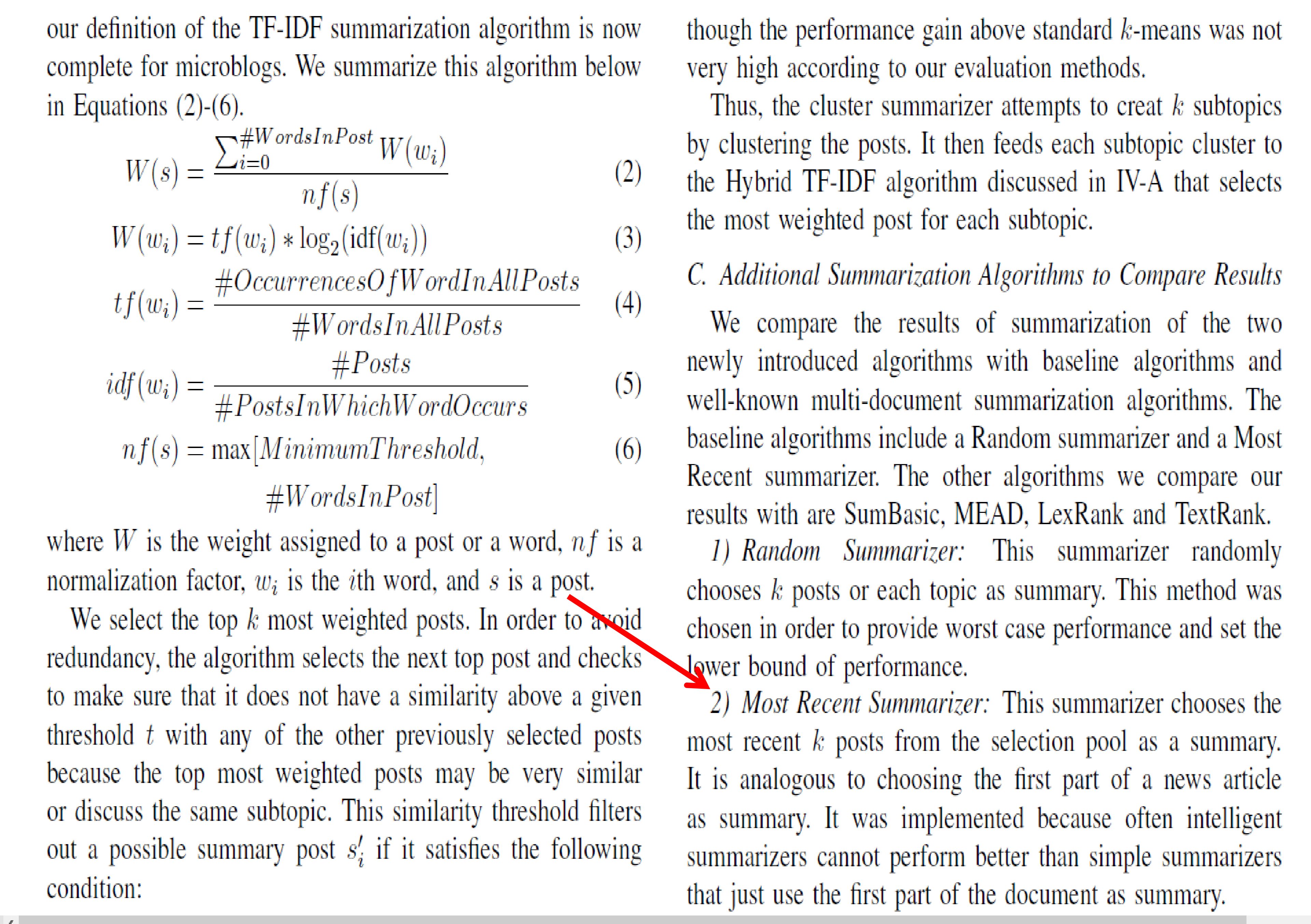
Thus, the cluster summarizer attempts to creat k subtopics by clustering the posts. It then feeds each subtopic cluster to the Hybrid TF-IDF algorithm discussed in IV-A that selects the most weighted post for each subtopic.

C. Additional Summarization Algorithms to Compare Results

We compare the results of summarization of the two newly introduced algorithms with baseline algorithms and well-known multi-document summarization algorithms. The baseline algorithms include a Random summarizer and a Most Recent summarizer. The other algorithms we compare our results with are SumBasic, MEAD, LexRank and TextRank.

1) Random Summarizer: This summarizer randomly chooses k posts or each topic as summary. This method was chosen in order to provide worst case performance and set the lower bound of performance.

2) Most Recent Summarizer: This summarizer chooses the most recent k posts from the selection pool as a summary. It is analogous to choosing the first part of a news article as summary. It was implemented because often intelligent summarizers cannot perform better than simple summarizers that just use the first part of the document as summary.



our definition of the TF-IDF summarization algorithm is now complete for microblogs. We summarize this algorithm below in Equations (2)-(6).

$$W(s) = \frac{\sum_{i=0}^{\#WordsInPost} W(w_i)}{nf(s)} \tag{2}$$

$$W(w_i) = tf(w_i) * \log_2(\mathrm{idf}(w_i)) \tag{3}$$

$$tf(w_i) = \frac{\#OccurrencesOfWordInAllPosts}{\#WordsInAllPosts} \tag{4}$$

$$idf(w_i) = \frac{\#Posts}{\#PostsInWhichWordOccurs} \tag{5}$$

$$nf(s) = \max[MinimumThreshold, \#WordsInPost] \tag{6}$$

where W is the weight assigned to a post or a word, nf is a normalization factor, w_i is the ith word, and s is a post.

We select the top k most weighted posts. In order to avoid redundancy, the algorithm selects the next top post and checks to make sure that it does not have a similarity above a given threshold t with any of the other previously selected posts because the top most weighted posts may be very similar or discuss the same subtopic. This similarity threshold filters out a possible summary post s'_i if it satisfies the following condition:

though the performance gain above standard k-means was not very high according to our evaluation methods.

Thus, the cluster summarizer attempts to creat k subtopics by clustering the posts. It then feeds each subtopic cluster to the Hybrid TF-IDF algorithm discussed in IV-A that selects the most weighted post for each subtopic.

C. Additional Summarization Algorithms to Compare Results

We compare the results of summarization of the two newly introduced algorithms with baseline algorithms and well-known multi-document summarization algorithms. The baseline algorithms include a Random summarizer and a Most Recent summarizer. The other algorithms we compare our results with are SumBasic, MEAD, LexRank and TextRank.

1) Random Summarizer: This summarizer randomly chooses k posts or each topic as summary. This method was chosen in order to provide worst case performance and set the lower bound of performance.

2) Most Recent Summarizer: This summarizer chooses the most recent k posts from the selection pool as a summary. It is analogous to choosing the first part of a news article as summary. It was implemented because often intelligent summarizers cannot perform better than simple summarizers that just use the first part of the document as summary.

3) SumBasic: SumBasic [5] uses simple word probabilities with an update function to compute the best k posts. It was chosen because it depends solely on the frequency of words in the original text and is conceptually very simple.

4) MEAD: This summarizer[5] [17] is a well-known flexible and extensible multi-document summarization system and was chosen to provide a comparison between the more structured document domain—in which MEAD works fairly well—and the domain of Twitter posts being studied. In addition, the default MEAD program is a cluster based summarizer so it will provide some comparison to our cluster summarizer.

5) LexRank: This summarizer [7] uses a graph based method that computes pairwise similarity between two sentences—in our case two posts—and makes the similarity score the weight of the edge between the two sentences. The final score of a sentence is computed based on the weights of the edges that are connected to it. This summarizer was chosen to provide a baseline for graph based summarization instead of direct frequency summarization. Though it does depend on frequency, this system uses the relationships among sentences to add more information and is therefore a more complex algorithm than the frequency based ones.

6) TextRank: This summarizer [8] is another graph based

SumBasic

- SumBasic’s underlying premise is that words that occur more frequently across documents have a higher probability of being selected for human created multi document summaries than words that occur less frequently.

V. Experimental Setup

A. Data Collection

For five consecutive days, we collected the top ten currently trending topics from Twitter's home page at roughly the same time every evening. For each topic, we downloaded the maximum number (approximately 1500) of posts. Therefore, we had 50 trending topics with a set of 1500 posts for each.

B. Preprocessing the Posts

Pre-processing steps included converting any Unicode characters into their ASCII equivalents, filtering out any embedded URL's, discarding spam using a Naïve Bayes classifier, etc. These pre-processing steps and their rationale are described more fully in [1].

TABLE III

AVERAGE VALUES OF F-MEASURE, RECALL AND PRECISION ORDERED BY F-MEASURE.

	F-measure	Recall	Precision
LexRank	0.2027	0.1894	0.2333
Random	0.2071	0.2283	0.1967
Mead	0.2204	0.3050	0.1771
Manual	0.2252	0.2320	0.2320
Cluster	0.2310	0.2554	0.2180
TextRank	0.2328	0.3053	0.1954
MostRecent	0.2329	0.2463	0.2253
Hybrid TF-IDF	0.2524	0.2666	0.2499
SumBasic	0.2544	0.3274	0.2127

significant variability about the best value for k. Our bias is also based on the fact that our initial 1500 Twitter posts on each topic were obtained within a small interval of 15 minutes so we thought a small number would be good.

Since the volunteers had already clustered the posts into four clusters, the manual summaries were four-post long as well. This kept the already onerous manual summary creation process somewhat simple. However, this also means that being dependent on a single length for the summaries may impact our evaluation process described next in an unknown way.

2) Manual Summarization Method: Our manual multi-post summaries were created by volunteers who were undergraduates from around the US gathered together in an NSF-supported REU program. Each of the first 25 topics was manually summarized by two different volunteers[7] by performing

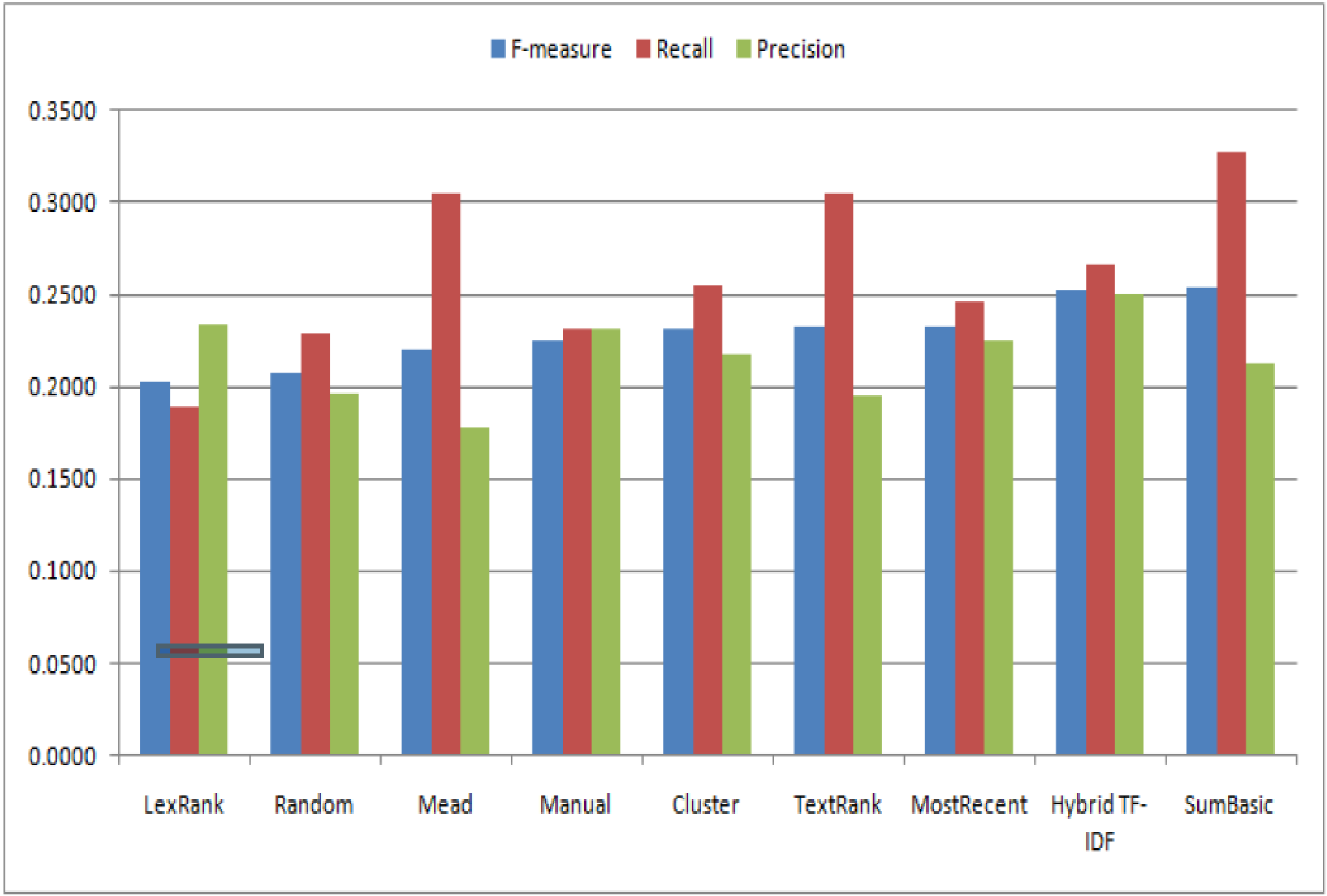


Fig. 2. Average F-measure, precision and recall ordered by F-measure.

Evaluation

- The evaluation measures for text summarization can be automatic or manual.
  - An example of automatic measure is ROUGE [Lin and Hovy, 2003] that compares the summary generated by the system with several manually written summaries.
  - It calculates n-gram overlap between the automatic summary and the multiple references (TP, FP), while penalizing for missed n-grams (FN). (See next slide for more details)

http://www.aclweb.org/anthology/W/W04/W04-1013.pdf

Evaluation: ROUGE

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing.

- The metrics compare an automatically produced summary or translation against a reference or a set of references (i.e. human-produced summary or translation).

https://en.wikipedia.org/wiki/ROUGE_(metric)

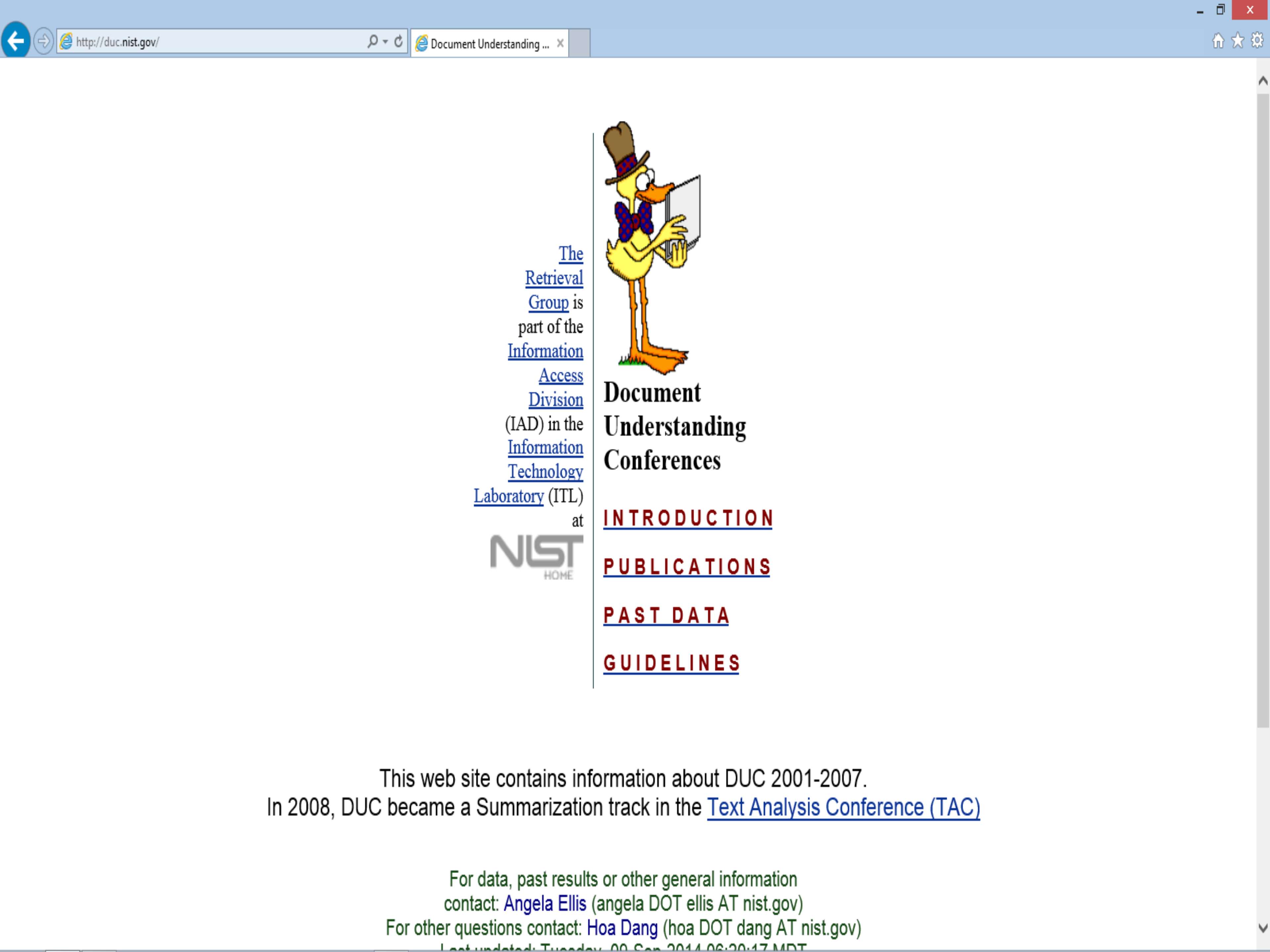
Evaluation: ROUGE

- The following five evaluation metrics are available:
    - ROUGE-N: N-gram based co-occurrence statistics.
    - ROUGE-L: Longest Common Subsequence (LCS) based statistics.
        - Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.
    - ROUGE-W: Weighted LCS-based statistics that favors consecutive LCSes .
    - ROUGE-S: Skip-bigram based co-occurrence statistics. Skip-bigram is any pair of words in their sentence order.
    - ROUGE-SU: Skip-bigram plus unigram-based co-occurrence statistics.

http://www.aclweb.org/anthology/W/W04/W04-1013.pdf

Evaluation: Manual

- The manual evaluation involves humans reading the generated summary and assessing its quality on several criteria.
    - **Responsiveness** measures the information content on a scale of 1 to 5.
    - **Readability** can also be on a scale of 1 to 5 (as an overall score, or for particular aspects such as grammaticality, non-redundancy, referential clarity, focus, structure, and coherence).
- These measures and a few others are used by the U.S. National Institute of Standards and Technology (NIST) in Document Understanding Conferences/Text Analysis Conferences (DUC/TAC). (See next slids)



The Retrieval Group is part of the Information Access Division (IAD) in the Information Technology Laboratory (ITL) at

Document Understanding Conferences

INTRODUCTION

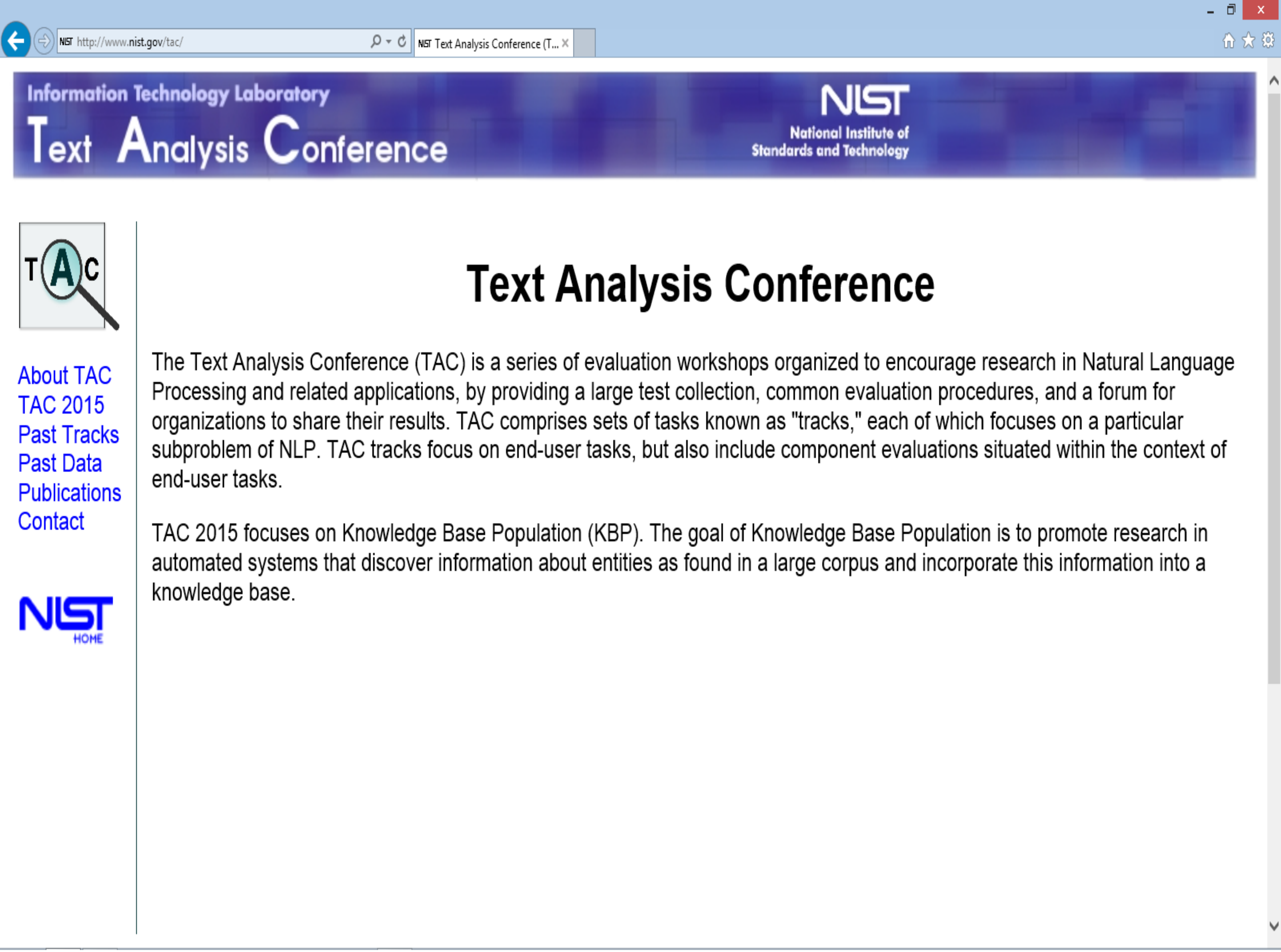
PUBLICATIONS

PAST DATA

GUIDELINES

This web site contains information about DUC 2001-2007.
In 2008, DUC became a Summarization track in the Text Analysis Conference (TAC)

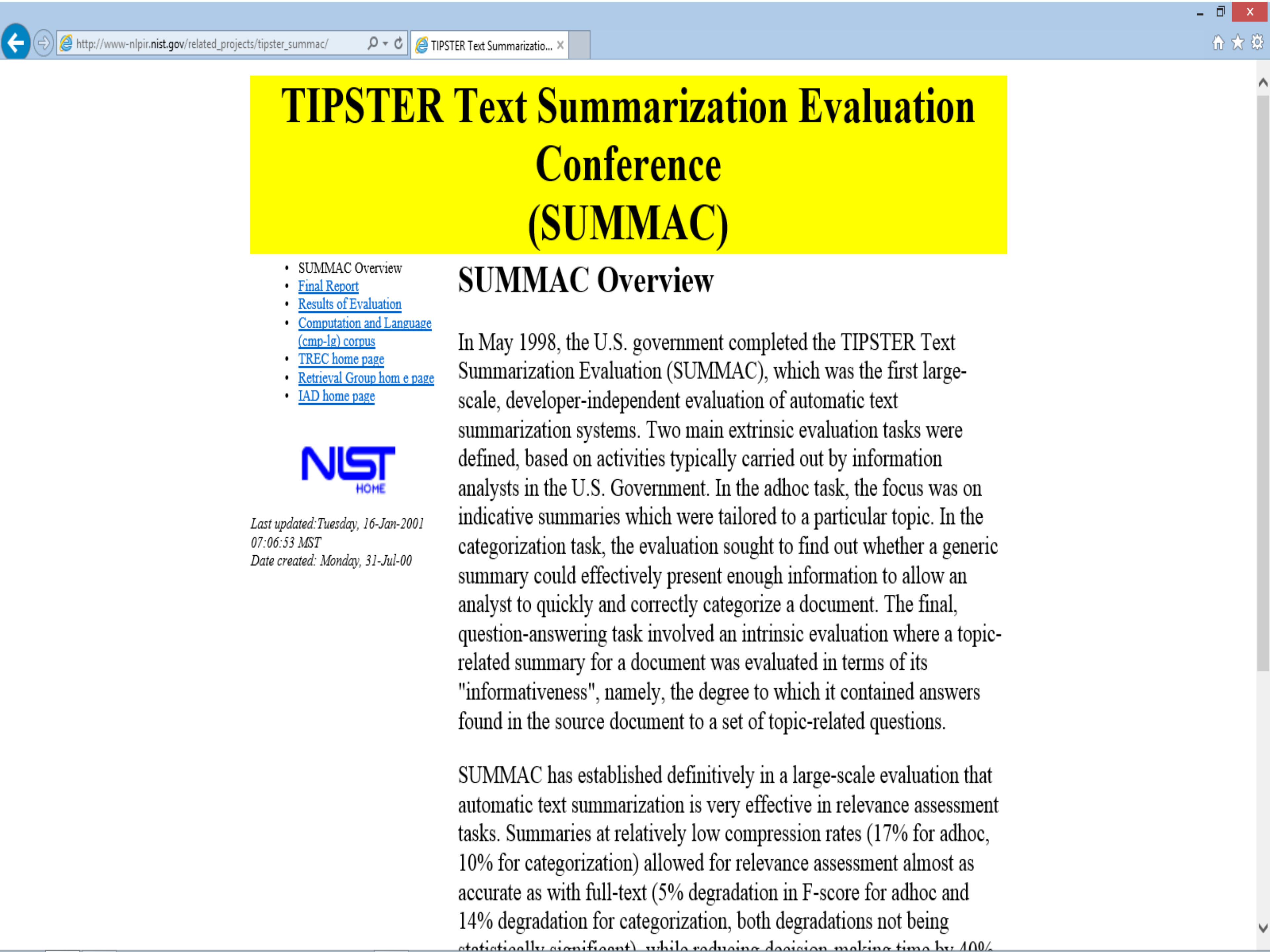
For data, past results or other general information
contact: Angela Ellis (angela DOT ellis AT nist.gov)
For other questions contact: Hoa Dang (hoa DOT dang AT nist.gov)
Last updated: Tuesday, 09-Sep-2014 06:20:17 MDT



Text Analysis Conference

The Text Analysis Conference (TAC) is a series of evaluation workshops organized to encourage research in Natural Language Processing and related applications, by providing a large test collection, common evaluation procedures, and a forum for organizations to share their results. TAC comprises sets of tasks known as "tracks," each of which focuses on a particular subproblem of NLP. TAC tracks focus on end-user tasks, but also include component evaluations situated within the context of end-user tasks.

TAC 2015 focuses on Knowledge Base Population (KBP). The goal of Knowledge Base Population is to promote research in automated systems that discover information about entities as found in a large corpus and incorporate this information into a knowledge base.


http://www-nlpir.nist.gov/related_projects/tipster_summac/
TIPSTER Text Summarizatio...

TIPSTER Text Summarization Evaluation Conference (SUMMAC)

- SUMMAC Overview
- Final Report
- Results of Evaluation
- Computation and Language (cmp-lg) corpus
- TREC home page
- Retrieval Group hom e page
- IAD home page

Last updated:Tuesday, 16-Jan-2001 07:06:53 MST
Date created: Monday, 31-Jul-00

SUMMAC Overview

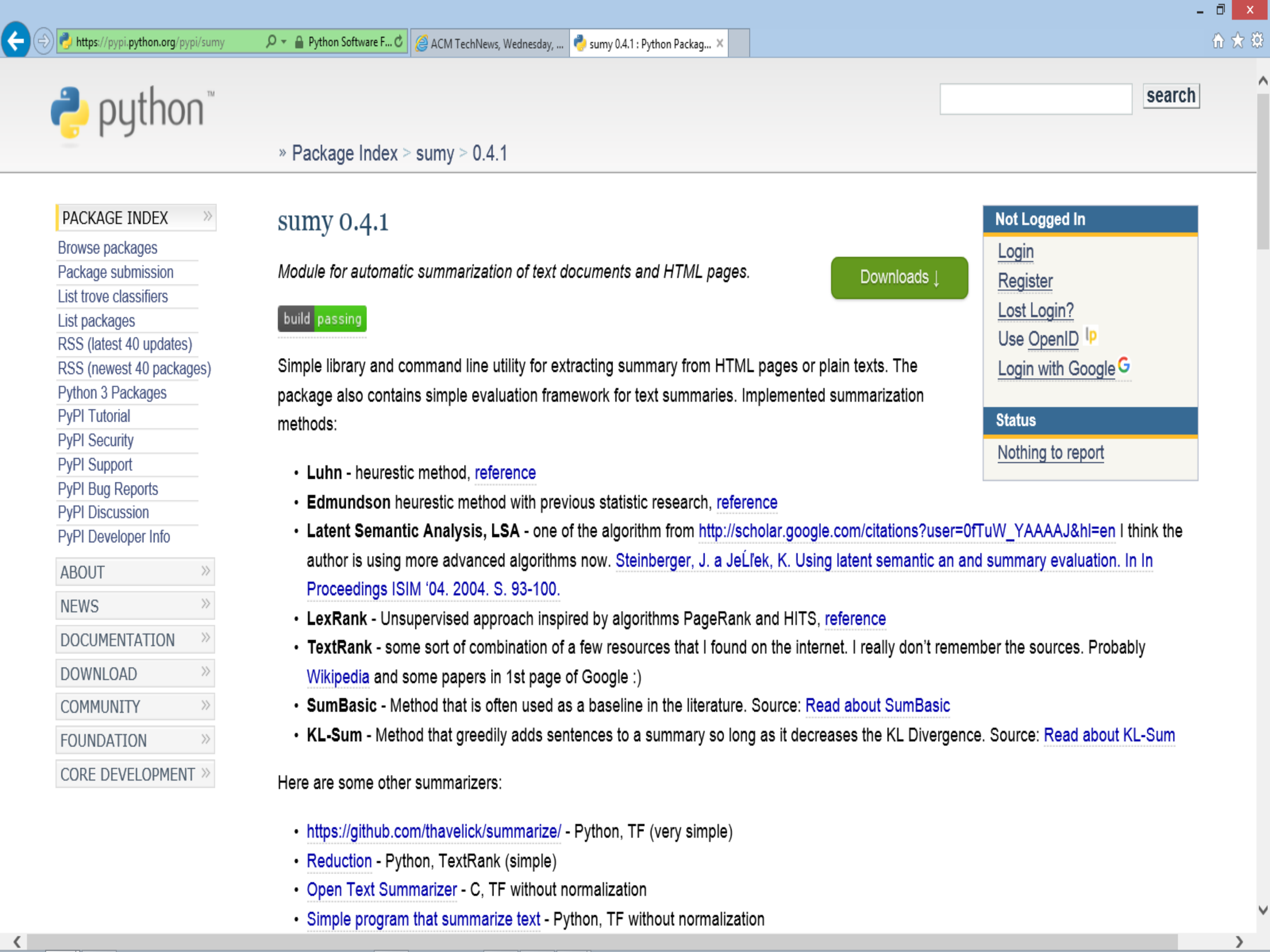
In May 1998, the U.S. government completed the TIPSTER Text Summarization Evaluation (SUMMAC), which was the first large-scale, developer-independent evaluation of automatic text summarization systems. Two main extrinsic evaluation tasks were defined, based on activities typically carried out by information analysts in the U.S. Government. In the adhoc task, the focus was on indicative summaries which were tailored to a particular topic. In the categorization task, the evaluation sought to find out whether a generic summary could effectively present enough information to allow an analyst to quickly and correctly categorize a document. The final, question-answering task involved an intrinsic evaluation where a topic-related summary for a document was evaluated in terms of its "informativeness", namely, the degree to which it contained answers found in the source document to a set of topic-related questions.

SUMMAC has established definitively in a large-scale evaluation that automatic text summarization is very effective in relevance assessment tasks. Summaries at relatively low compression rates (17% for adhoc, 10% for categorization) allowed for relevance assessment almost as accurate as with full-text (5% degradation in F-score for adhoc and 14% degradation for categorization, both degradations not being statistically significant), while reducing decision-making time by 40%

Evaluation: ROUGE

- Specifically for microblog summarization, Mackie et al. [2014] showed that a new metric, **the fraction of topic words** found in the summary, better agrees with what users perceive about the quality and effectiveness of microblog summaries than the ROUGE measure that is most commonly reported in the literature.

Reference:
Stuart Mackie, Richard McCreadie, Craig Macdonald, and Iadh Ounis. On choosing an effective automatic evaluation metric for microblog summarisation. In *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX ’14, pages 115–124, New York, NY, USA, 2014. ACM. DOI: 10.1145/2637002.2637017. 66


https://pypi.python.org/pypi/sumy
Python Software F...
ACM TechNews, Wednesday, ...
sumy 0.4.1 : Python Packag...
python™
search

» Package Index > sumy > 0.4.1

sumy 0.4.1

Module for automatic summarization of text documents and HTML pages.

Simple library and command line utility for extracting summary from HTML pages or plain texts. The package also contains simple evaluation framework for text summaries. Implemented summarization methods:

- **Luhn** - heurestic method, reference
- **Edmundson** heurestic method with previous statistic research, reference
- **Latent Semantic Analysis, LSA** - one of the algorithm from http://scholar.google.com/citations?user=0fTuW_YAAAAJ&hl=en I think the author is using more advanced algorithms now. Steinberger, J. a JeĹľek, K. Using latent semantic an and summary evaluation. In In Proceedings ISIM '04. 2004. S. 93-100.
- **LexRank** - Unsupervised approach inspired by algorithms PageRank and HITS, reference
- **TextRank** - some sort of combination of a few resources that I found on the internet. I really don't remember the sources. Probably Wikipedia and some papers in 1st page of Google :)
- **SumBasic** - Method that is often used as a baseline in the literature. Source: Read about SumBasic
- **KL-Sum** - Method that greedily adds sentences to a summary so long as it decreases the KL Divergence. Source: Read about KL-Sum

Here are some other summarizers:

- https://github.com/thavelick/summarize/ - Python, TF (very simple)
- Reduction - Python, TextRank (simple)
- Open Text Summarizer - C, TF without normalization
- Simple program that summarize text - Python, TF without normalization

Latent Semantic Analysis

- Latent semantic analysis (LSA) is a unsupervised technique for deriving an <u>implicit</u> representation of text semantics based on observed co-occurrence of words.

  S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, pages 391–407, 1990.

- [Gong & Liu, 2001] proposed the use of LSA for single and multi-document generic summarization of news, as a way of identifying important topics in documents without the use of lexical resources such as **WordNet**.

  Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. *Proceedings of the Annual International ACM SIGIR*, pages 19–25,2001.

Latent Semantic Analysis

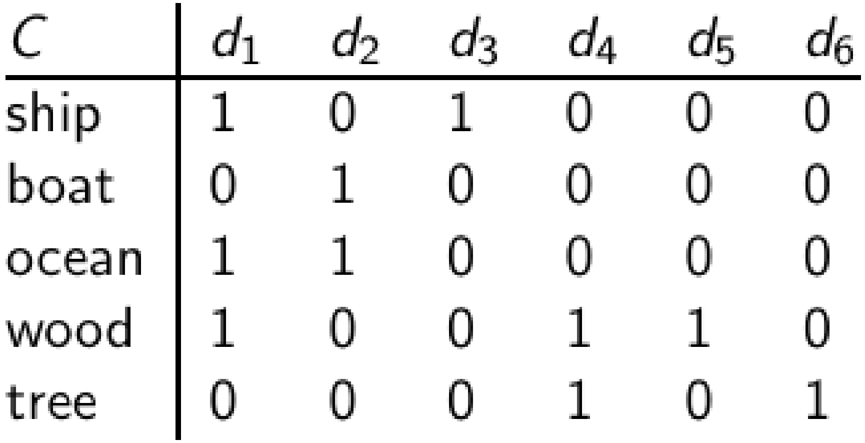
- Building the topic representation starts by filling in a ***n*** by ***m*** matrix *C*: each row corresponds to a word from the input (*n* words) and each column corresponds to a sentence in the input (*m* sentences).
  - Entry $\boldsymbol{c_{ij}}$ of the matrix corresponds to the weight of word *i* in sentence *j*. ‘
  - If the sentence does not contain the word, the weight is zero, otherwise the weight is equal to the **tf-idf** weight of the word.
- Standard techniques for singular value decomposition (**SVD**) from linear algebra are applied to the matrix *A*, to represent it as the product of three matrices: $C = U\Sigma V^T$.
  - Every matrix has a representation of this kind and many standard libraries provide a built-in implementation of the decomposition

Latent Semantic Analysis

- Matrix U is a n by m matrix of real numbers.
  - Each column can be interpreted as a **topic**, i.e. a specific combination of words from the input with the weight of each word in the topic given by the real number.
- Matrix Σ is diagonal m by m matrix.
  - The single entry in row i of the matrix corresponds to the weight of the **topic**, which is the i_{th} column of U.
  - Topics with low weight can be ignored, by deleting the last k columns of U, the last k rows and columns of Σ and the last k rows of V^T .
  - This procedure is called **dimensionality reduction**.
- Matrix V^T is a new representation of the sentences, one sentence per row, each of which is expressed not in terms of words that occur in the sentence but rather in terms of the topics given in U.
- The matrix $D = \Sigma V^T$ combines the topic weights and the sentence representation to indicate to what extent the sentence conveys the topic, with d_{ij} indicating the weight for topic i in sentence j.

LSA: A Simpler Example

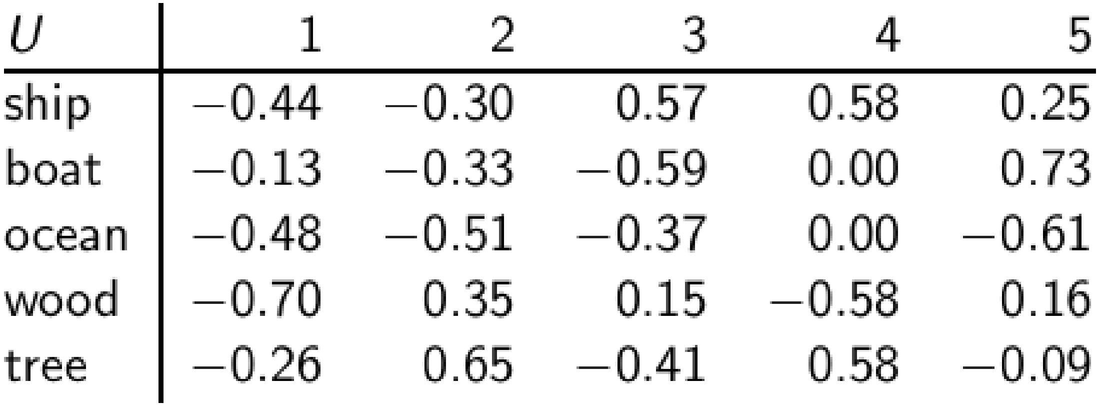
Example of $C = U\Sigma V^T$: The matrix C



C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

LSA: A Simpler Example

Example of $C = U\Sigma V^T$: The matrix U

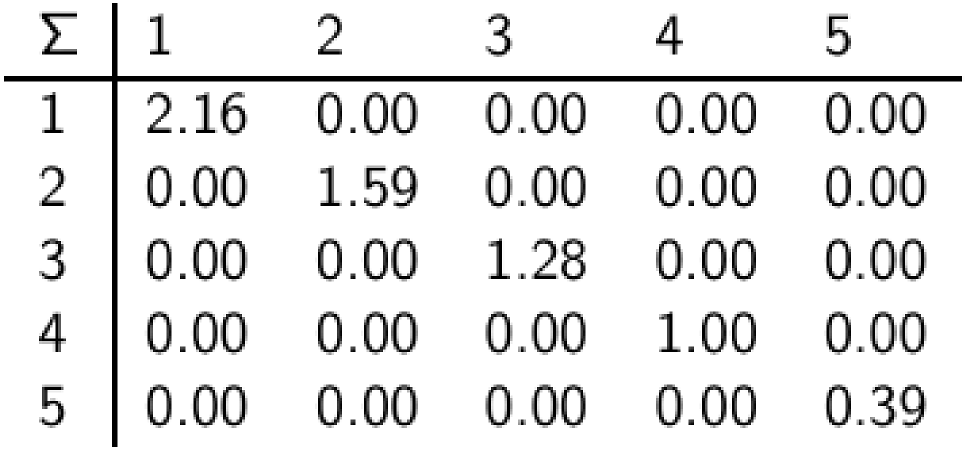


U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

This is an orthonormal matrix:
(i) Row vectors have unit length. (ii) Any two distinct row vectors are orthogonal to each other. Think of the dimensions as **semantic** dimensions that capture distinct topics like politics, sports, economics. Each number u_{ij} in the matrix indicates how strongly related term i is to the topic represented by semantic dimension j .

LSA: A Simple Example

Example of $A = U\Sigma V^T$: The matrix Σ

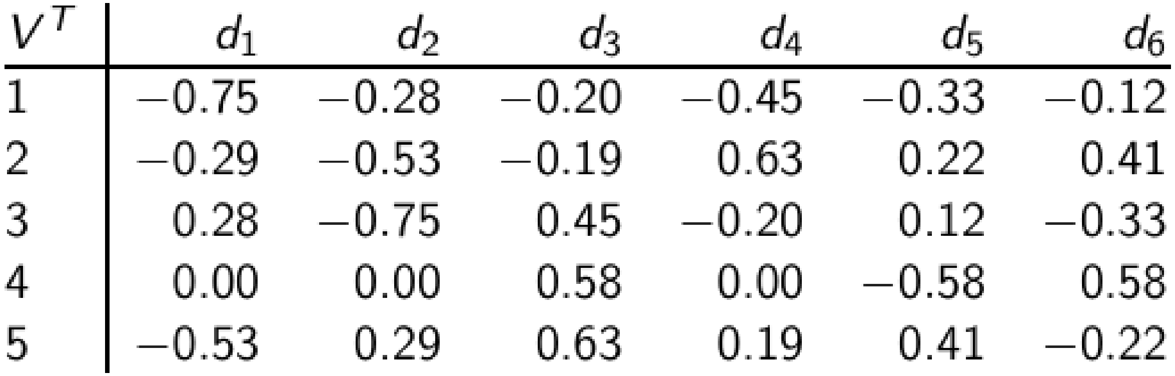


Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

This is a square, diagonal matrix of dimensionality min(m,n) × min(m,n). The diagonal consists of the singular values of C. The magnitude of the singular value measures the importance of the corresponding semantic dimension. We can make use of this by omitting unimportant dimensions.

LSA: A Simple Example

Example of $C = U\Sigma V^T$: The matrix V^T

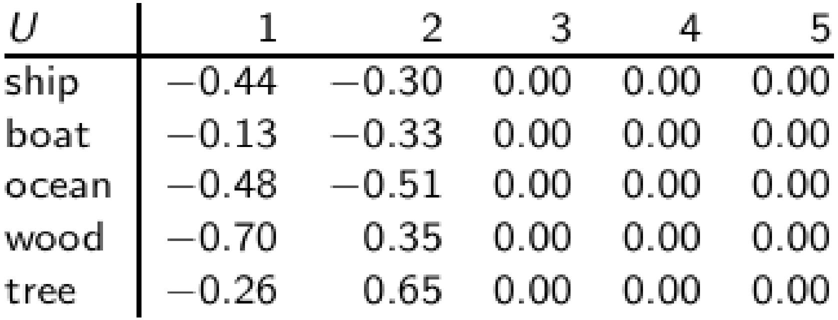


V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

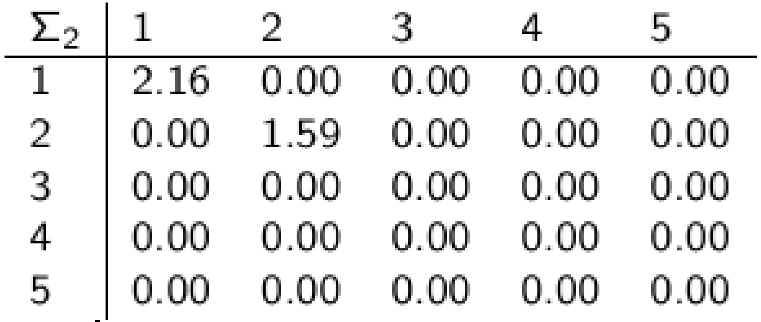
One column per document, one row per min(m,n) where m is the number of terms and n is the number of documents/sentences. Again: This is an orthonormal matrix: (i) Column vectors have unit length. (ii) Any two distinct column vectors are orthogonal to each other. These are again the semantic dimensions from the term matrix U that capture distinct topics like politics, sports, economics. Each number v_{ij} in the matrix indicates how strongly related document i is to the topic represented by semantic dimension j .

LSA: A Simple Example

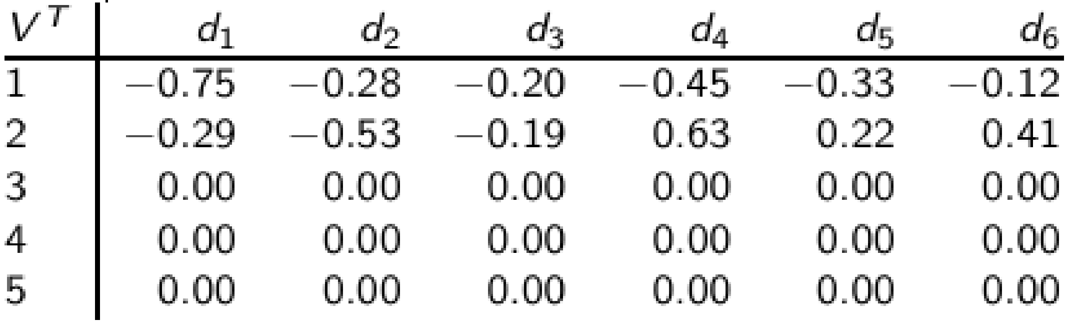
Reducing the dimensionality to 2



U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00



Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00



V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Actually, we only zero out singular values in Σ. This has the effect of setting the corresponding dimensions in *U* and *V* *T* to zero when computing the Product $C = U\Sigma V^T$.

Latent Semantic Analysis

- Matrix U is a n by m matrix of real numbers.
    - Each column can be interpreted as a **topic**, i.e. a specific combination of words from the input with the weight of each word in the topic given by the real number.
- Matrix Σ is diagonal m by m matrix.
    - The single entry in row i of the matrix corresponds to the weight of the **topic**, which is the i_{th} column of U.
    - Topics with low weight can be ignored, by deleting the last k columns of U, the last k rows and columns of Σ and the last k rows of V^T .
    - This procedure is called **dimensionality reduction**.
- Matrix V^T is a new representation of the sentences, one sentence per row, each of which is expressed not in terms of words that occur in the sentence but rather in terms of the topics given in U.
- **The matrix $D = \Sigma V^T$ combines the topic weights and the sentence representation to indicate to what extent the sentence conveys the topic, with d_{ij} indicating the weight for topic i in sentence j.**

Latent Semantic Analysis

- The original proposal of Gong and Liu was to select one sentence for each of the most important topics.
  - They perform dimensionality reduction, retaining only as many topics as the number of sentences they want to include in the summary.
  - The sentence with the highest weight for each of the retained topics is selected to form the summary.
- The drawback of this strategy is that more than one sentence may be required to convey all information pertinent to that topic.

Latent Semantic Analysis

- Researchers have proposed alternative procedures which have led to improved performance of the summarizer in content selection.
  - One improvement is to use the weight of each topic in order to determine the relative proportion of the summary that should cover the topic, thus allowing for a variable number of sentences per topic.
  - Another improvement was to notice that often sentences that discuss several of the important topics are good candidates for summaries.

J. Steinberger, M. Poesio, M. A. Kabadjov, and K. Jeek. Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43(6):1663–1680, 2007.

In general the KL divergence of probability distribution Q with respect to distribution P over words w is defined as

$$KL(P||Q) = \sum_{w} P(w) \log \frac{P(w)}{Q(w)} \tag{3.7}$$

$P(w)$ and $Q(w)$ are the probabilities of w in P and Q respectively.

Sentences are scored and selected in a greedy iterative procedure [36]. In each iteration the best sentence i to be selected in the summary is determined as the one for which the KL divergence between C, the probabilities of words in the cluster to be summarized, and the summary so far, including i, is smallest.

KL divergence is appealing as a way of scoring and selecting sentence in summarization because it truly captures an intuitive notion that good summaries are similar to the input. Thinking about a good summary in this way is not new in summarization [21, 74] but KL provides a way of measuring how the importance of words, given by their probabilities, changes in the summary compared to the input. A good summary would reflect the importance of words according to the input, so the divergence