A Clustering Based Approach to Creating Multi-Document Summaries

Endre Boros[a], Paul B. Kantor[a,b] and David J. Neu[a]
[a]RUTCOR, Rutgers University
[b]SCILS, Rutgers University
{boros,neu}@rutcor.rutgers.edu, kantor@scils.rutgers.edu

August 24, 2001

Abstract

Multi-document summaries were constructed by utilizing *complete* sentences from the documents in the collection. Classic clustering techniques were employed in an attempt to partition the set of sentences into disjoint subsets or clusters, each of which contained sentences covering exactly one topic. Clusters are ranked by their similarity with the vector of the term frequencies of all terms appearing in the documents to be summarized. While the system performance appears to be poor, we believe that the method warrants future research.

1 Introduction

In this, the first year that the Rutgers University team participated in the Document Understanding Conference, we participated in the *multi-document summarization* task.

In the multi-document summarization task, thirty document sets, each on a different subject, and each consisting of approximately ten documents are provided as input. The goal of the task is, for the documents in each set, to create four summaries, containing no more than 400, 200, 100, and 50 words respectively.

To accomplish this task we assumed that:

1. The *subject* of the documents in each set could be viewed as consisting of a finite number of "topics", and that the desired summary should cover as many of these topics as the length constraints permitted.

2. The topics for each subject could be ordered by how "important" it is that they be covered in a summary.

In order to employ these assumptions to create the desired summaries:

1. Each complete sentence in an original document is considered to cover exactly one topic.

2. The summaries will consist of complete sentences from the original documents.

2 System

The system performed the following steps:

Parse Documents Into Sentences: Documents were parsed into sentences using the end of sentence punctuation (i.e. ".", "!", "?") as markers except in cases in which the "." was immediately preceded by an abbreviation. An abbreviation list was constructed by manually inspecting all strings in the trial set which began with a capital letter and were immediately followed by a ".". Only sentences with between five and thirty words are used. The rational was that sentences with more than thirty word probably would not get be used by the summary generation procedure due to length limitations and that sentences with less than five words probably do not carry much information.

INPUT: Original SGML documents.
OUTPUT: `$TOPIC-sentence.txt`
FORMAT: documentID sentenceID sentenceString

Create Sentence Term Index: A sentence term index was created. The terms in this index did not include stopwords (as listed on the SMART stopword list), abbreviations and terms containing numbers.

INPUT: `$TOPIC-sentence.txt`
OUTPUT: `$TOPIC-sentenceIndex.txt`
FORMAT: documentID sentenceID term frequencyWithinSentence

Create Term Frequency Index for the Subject Documents: A standard relative term frequency index was created for the terms appearing in the documents to be summarized. This information was used to create the *collection term frequency vector* which is viewed as the "collection center".

INPUT: `$TOPIC-sentenceIndex.txt`
OUTPUT: `$TOPIC-collectionIndex.txt`
FORMAT: term frequencyWithinCollection

Calculate Percentage of Sentences In Which Each Term is Used: A file which maintains the percentage of sentences in which each term contained in the documents to be summarized is used was created. This information was used during cluster creation to reduce the dimensionality of the computations.

INPUT: `$TOPIC-sentenceIndex.txt`
OUTPUT: `$TOPIC-collectionSentenceIndex.txt`
FORMAT: term percentOfSentencesInWhichTermIsUsed

Create Clusters of Sentences: Using classical clustering methods, the set of sentences were partitioned into disjoint subsets or clusters. Two files were created. The first file indicates which sentences belong to which clusters. In addition, it includes the distance from each sentence to its cluster center, the sentence length and the sentence position in the original document. The second file provides statistics about each cluster, including the distance from the cluster center to the collection center and the average position of the sentences contained in the cluster.

INPUT: $TOPIC-sentence.txt, $TOPIC-sentenceIndex.txt, $TOPIC-collectionIndex.txt, $TOPIC-collectionSentenceIndex.txt
OUTPUT: $TOPIC-cluster.txt
FORMAT: `clusterID documentID-sentenceID distToClusterCenter sentenceLength sentencePos`
OUTPUT: $TOPIC-cluster-center.txt
FORMAT: `clusterID distToCollectionCenter clusterSentencePosAvg`

The specific clustering method used was a combination of *hierarchical* and *non-hierarchical* methods:

- Hierarchical clustering was used for finding the initial clusters:
  1. Start with each sentence being a cluster of size 1.
  2. Calculate the distance between each cluster and sort a list of this information so the "closest" clusters are at the top.
  3. Pick the two clusters which are "closest" and merge them into a new cluster.
  4. Delete the two "closest" clusters and any references to them in the distance list.
  5. Go to 2.
  6. Stop when have trimmed down to 30 clusters.
- Non-non-hierarchical clustering, specifically *k-means* is given the 30 clusters as a starting point, with a target of trimming the number of clusters to 10. Since k-means may terminate with more than the target of 10 clusters, the 10 clusters with the most sentence in them are utilized.

In both steps the distance measure employed is the difference between 1.0 and the cosine similarity measure.

Create Summary: To determine which sentences should be selected to be included in the summary and the order in which they should appear, clusters

were ranked by their similarity (using the cosine similarity measure) to the collection term frequency vector. The sentences within each cluster were then ranked by their similarity to their cluster center. Iterating over the ordered clusters, and over the ordered sentences within each cluster, one sentence at each iteration was selected to be included in the summary.

3 Evaluation

Evaluation concentrated on coverage of model units by peer units. We compute the average coverage of model units, given the data sent by NIST. Then, separately, for each size of summary, we compute the rank of each system, for each of the topics. From this we compute an average rank. (A Freidman type statistic). Significance intervals are known, and can be estimated presuming that ranks are distributed uniformly. Application suggests that our own system performs very poorly.

4 Conclusion

While our preliminary analysis seems to indicate that our system performed poorly, we still believe that the method is promising and that subsequent refinement may yield improved results.

Areas for future research:

- Investigation of alternative methods for determining the number of clusters.
- Investigation of methods other than classical clustering algorithms for topic detection.
- Since anecdotal evidence indicates that pruning very short and very long sentences increases performance, it seems that further investigation of methods for removing "noise" sentences is warranted.
- Enhancement of our sentence boundary identification tools.

References

[1] Michael R. Anderberg. *Cluster Analysis for Applications.* Academic Press, 1973.

[2] Michael W. Berry and Murray Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval.* SIAM, 1999.

[3] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

Notes

A Clustering Based Approach to Creating Multi-Document Summaries

Endre Boros, Paul B. Kantor, David J. Neu

Rutgers University

Assumptions:

1. Each document collection is considered to consists of a small finite number of “themes”
2. A “good” summary should cover as many of these themes as length constraints permit
3. Themes can be ordered by how “important” it is that they be covered in the summary
4. Each complete sentence in a collection document, is considered to cover at most one “theme”

Approach:

Summaries will consist of complete sentences selected from collection documents

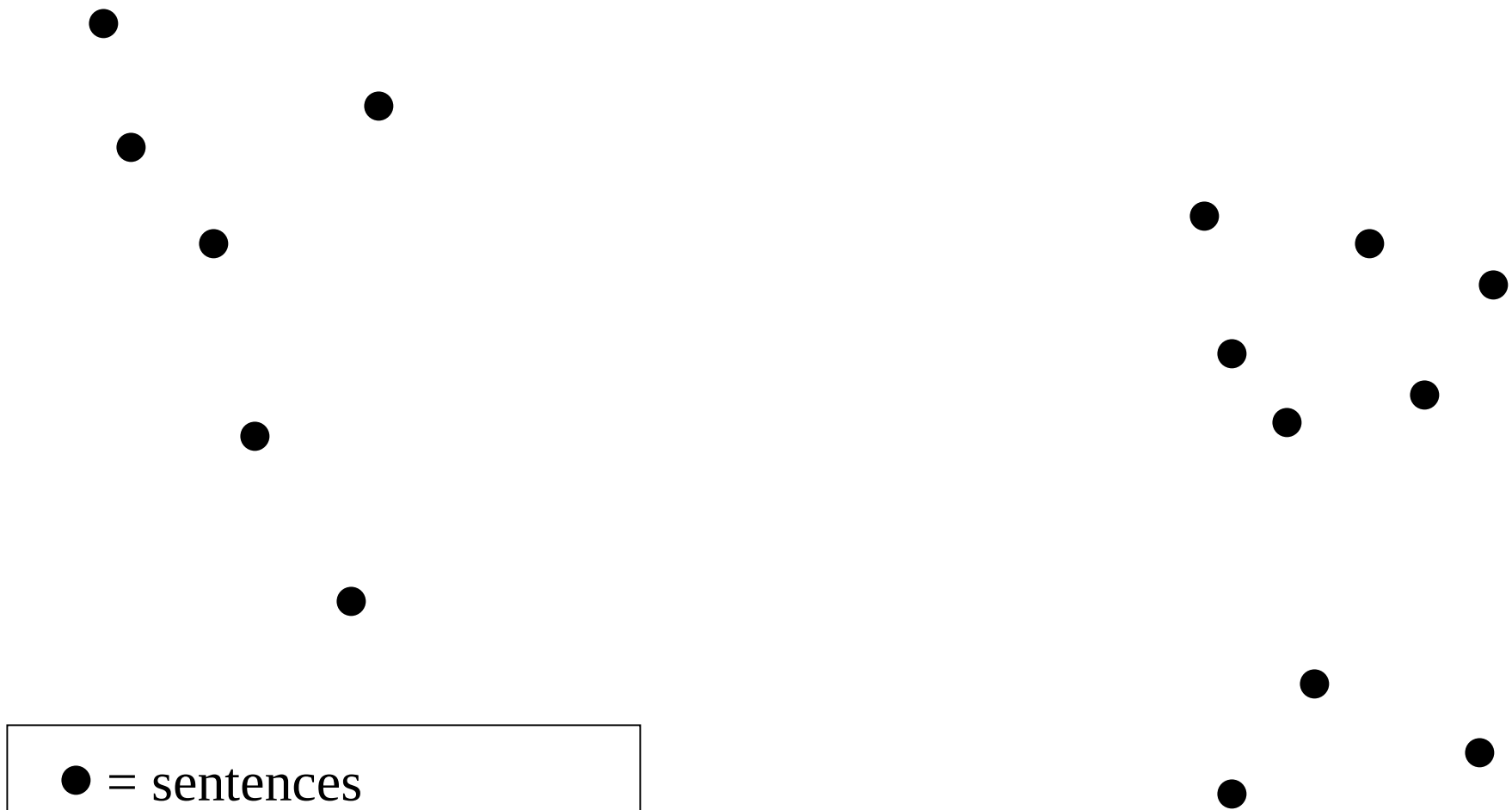
Manually Selected “Themes” from d01: Clarence Thomas

- -nominated to the Supreme Court on July 1, 1991
- -conservative
- -republican
- -controversial civil rights record
- -second African-American to be appointed
- -nominated by President Bush
- -position on affirmative action
- -position on abortion
- -Roman Catholic
- -Anita Hill accused him of sexual harassment
- -contentious confirmation process
- -confirmed on October 16, 1991

Model

1. Parse the document collection into a set of sentences.
2. Utilize classic (unsupervised) cluster analysis techniques to partition the set of sentences into theme clusters, i.e. disjoint subsets of sentences, such that each sentence in a cluster is "about" the same theme.
3. Compute the cluster centers as $(\mu_1, \ldots, \mu_n)$, where μ_i is the average frequency of the i^{tih} term.
4. For each cluster, compute the distance from each sentence *s*, to its cluster center *c*, as *1-cos(s,c)*.
5. Consider the document collection center (modeling what the collection is "about") to be the term frequency vector of the entire collection.
6. Compute the distance from each cluster center *c*, to the document collection center *d* as *1-cos(d,c)*.

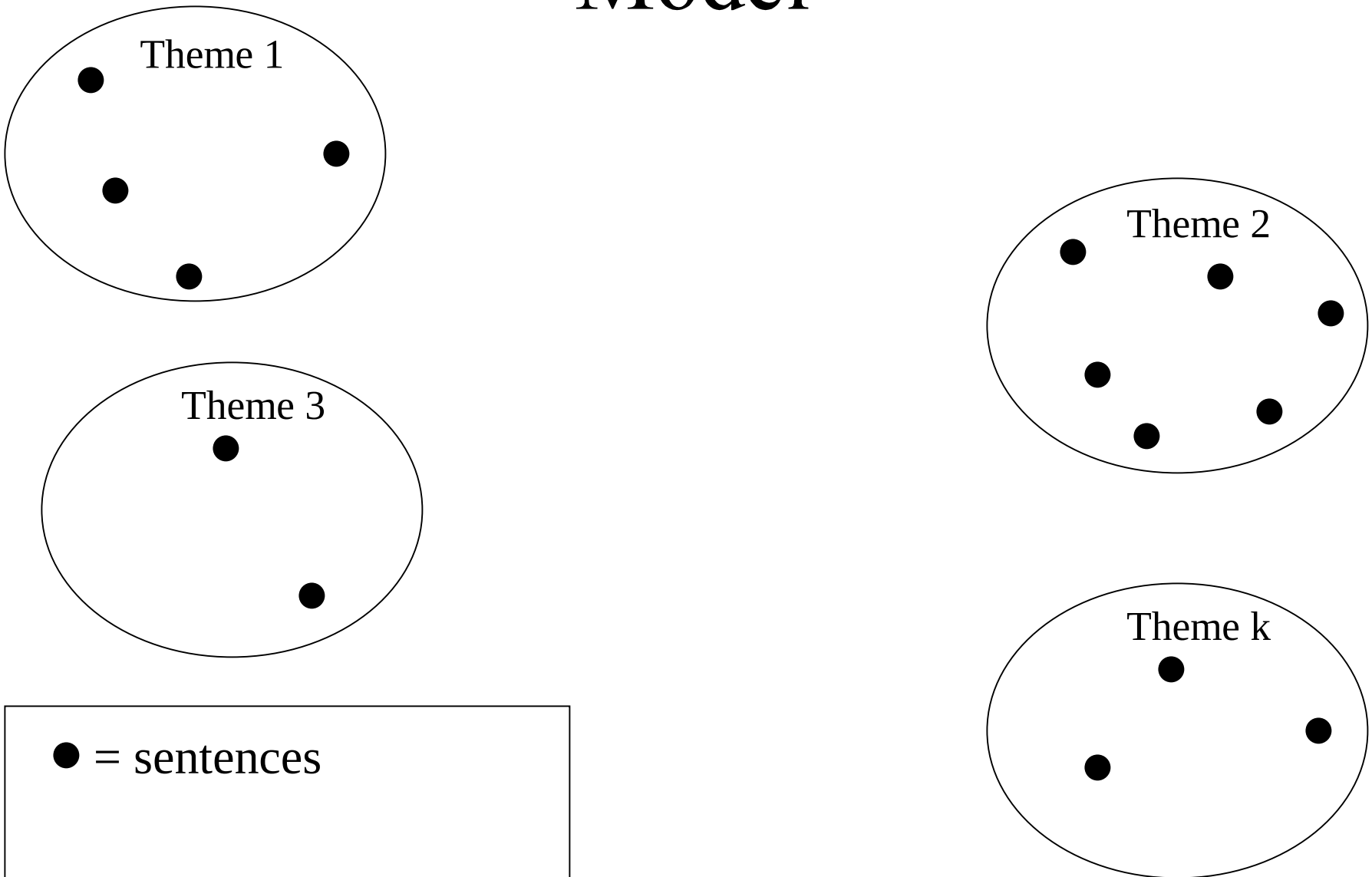
Model



Model

1. Parse the document collection into a set of sentences.
2. Utilize classic (unsupervised, seeded) cluster analysis techniques to partition the set of sentences into theme clusters, i.e. disjoint subsets of sentences, such that each sentence in a cluster is "about" the same theme.
3. Compute the cluster centers as $(\mu_1, \ldots, \mu_n)$, where μ_i is the average frequency of the i^{tih} term.
4. For each cluster, compute the distance from each sentence *s*, to its cluster center *c*, as *1-cos(s,c)*.
5. Consider the document collection center (modeling what the collection is "about") to be the term frequency vector of the entire collection.
6. Compute the distance from each cluster center *c*, to the document collection center *d* as *1-cos(d,c)*.

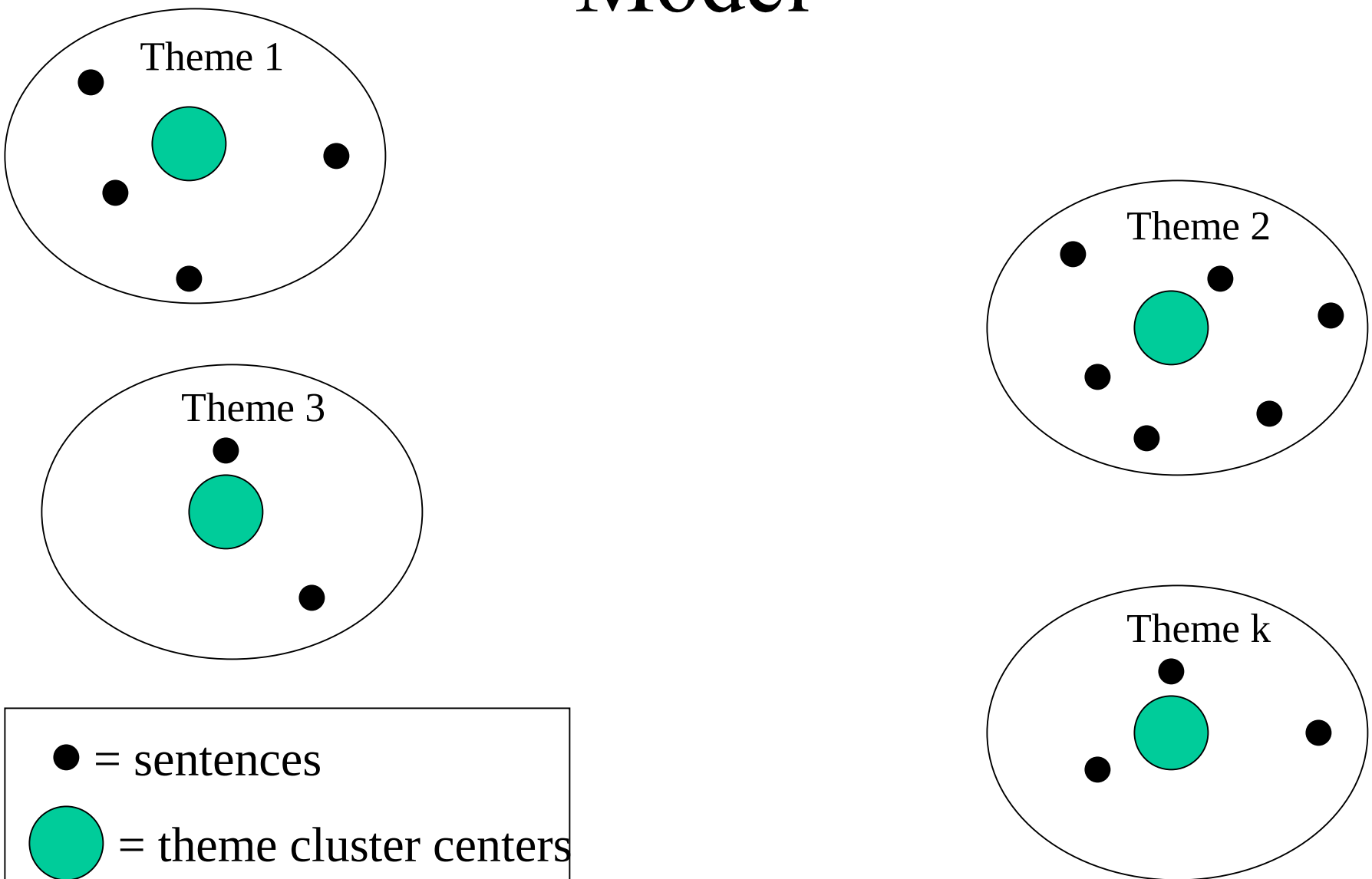
Model



Model

1. Parse the document collection into a set of sentences.
2. Utilize classic (unsupervised) cluster analysis techniques to partition the set of sentences into theme clusters, i.e. disjoint subsets of sentences, such that each sentence in a cluster is "about" the same theme.
3. Compute the cluster centers as $(\mu_1, \ldots, \mu_n)$, where μ_i is the average frequency of the i^{th} term.
4. For each cluster, compute the distance from each sentence *s*, to its cluster center *c*, as *1-cos(s,c)*.
5. Consider the document collection center (modeling what the collection is "about") to be the term frequency vector of the entire collection.
6. Compute the distance from each cluster center *c*, to the document collection center *d* as *1-cos(d,c)*.

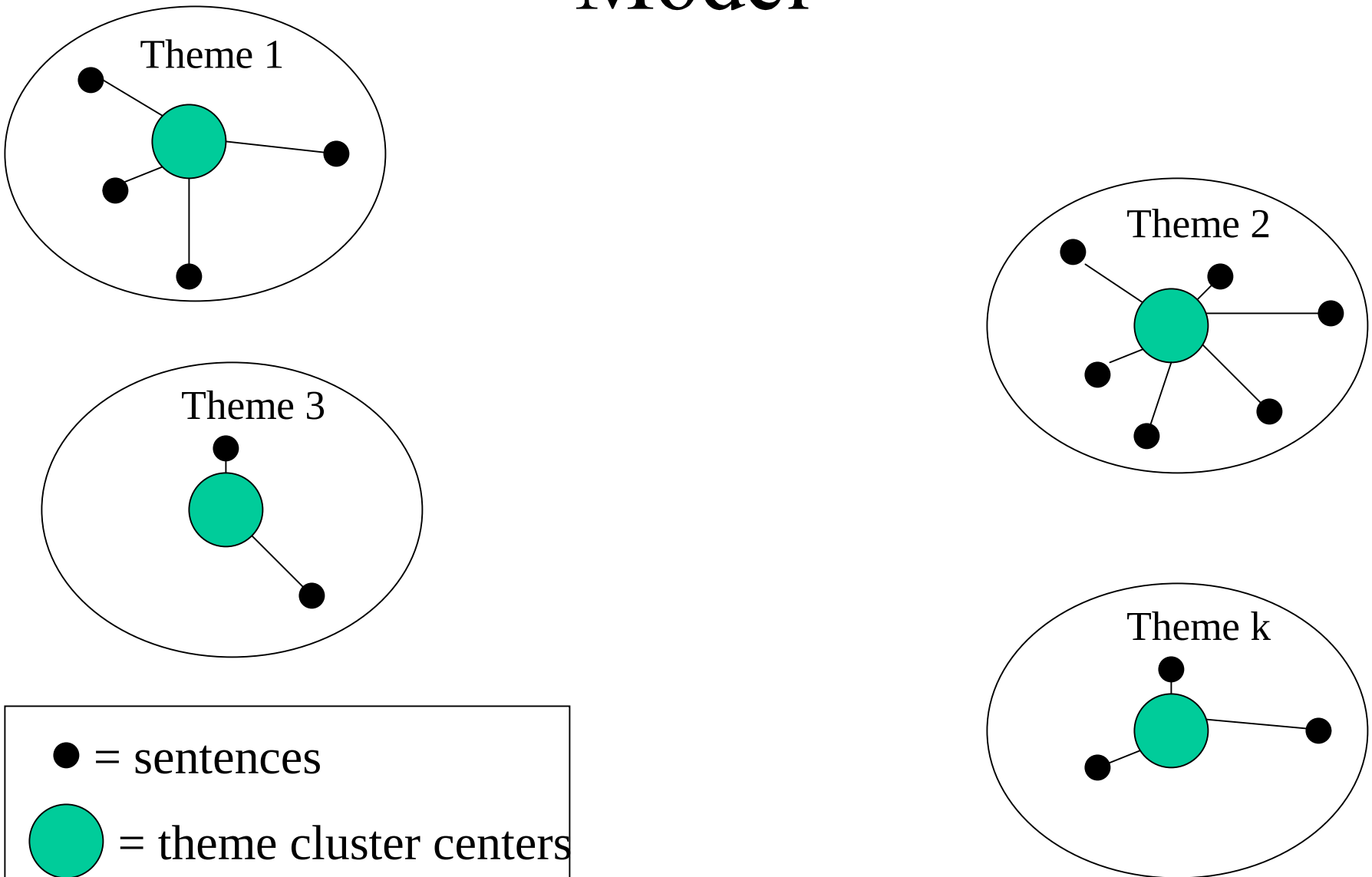
Model


Theme 1
Theme 2
Theme 3
Theme k
= sentences
= theme cluster centers

Model

1. Parse the document collection into a set of sentences.
2. Utilize classic (unsupervised) cluster analysis techniques to partition the set of sentences into theme clusters, i.e. disjoint subsets of sentences, such that each sentence in a cluster is "about" the same theme.
3. Compute the cluster centers as $(\mu_1, \ldots, \mu_n)$, where μ_i is the average frequency of the i^{th} term.
4. For each cluster, compute the distance from each sentence *s*, to its cluster center *c*, as *1-cos(s,c)*.
5. Consider the document collection center (modeling what the collection is "about") to be the term frequency vector of the entire collection.
6. Compute the distance from each cluster center *c*, to the document collection center *d* as *1-cos(d,c)*.

Model


Theme 1
Theme 2
Theme 3
Theme k
= sentences
= theme cluster centers

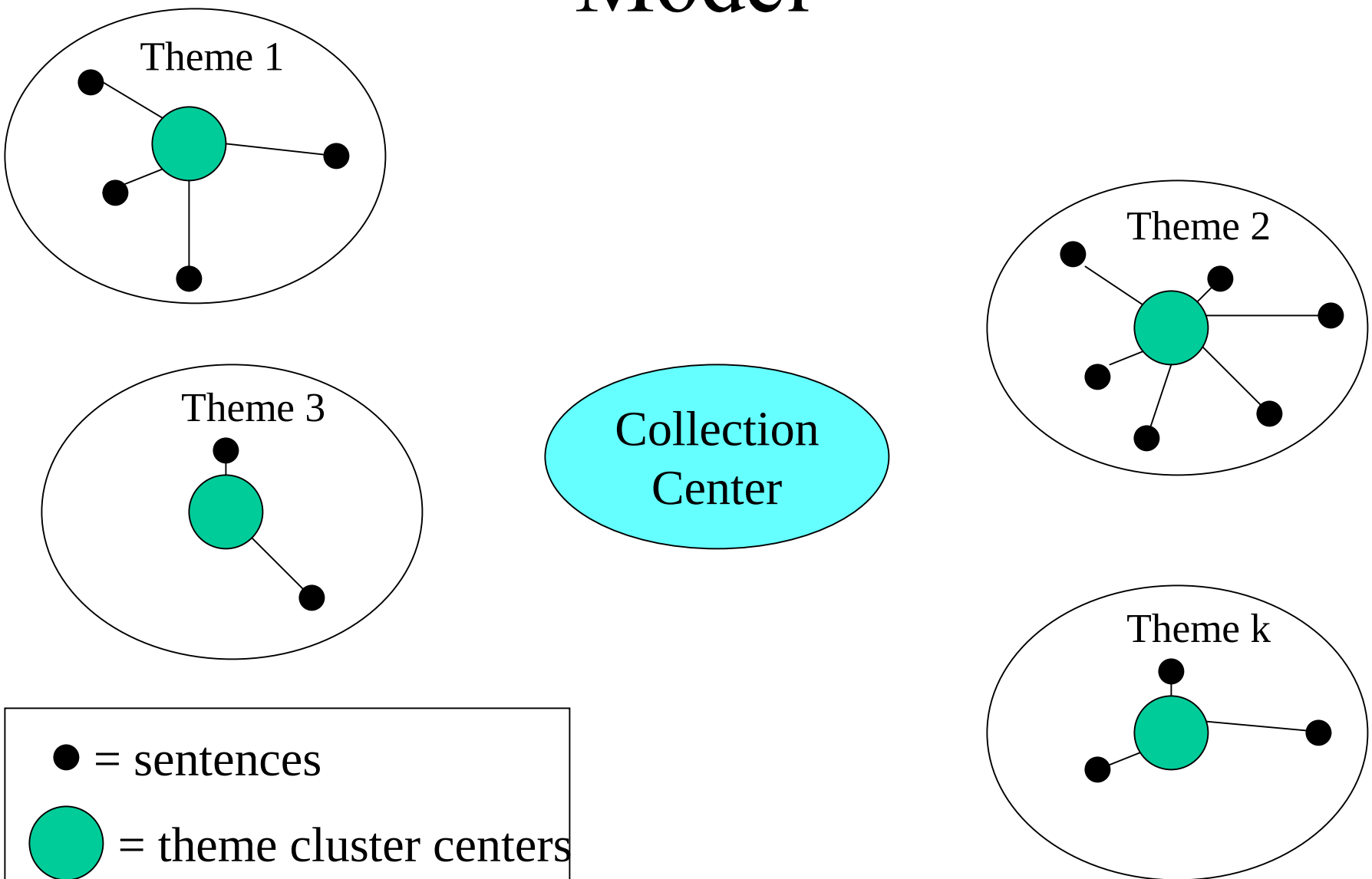
Model

1. Parse the document collection into a set of sentences.
2. Utilize classic (unsupervised) cluster analysis techniques to partition the set of sentences into theme clusters, i.e. disjoint subsets of sentences, such that each sentence in a cluster is "about" the same theme.
3. Compute the cluster centers as $(\mu_1, \ldots, \mu_n)$, where μ_i is the average frequency of the i^{th} term.
4. For each cluster, compute the distance from each sentence *s*, to its cluster center *c*, as *1-cos(s,c)*.
5. Consider the document collection center (modeling what the entire collection is "about") to be the term frequency vector of the entire collection.
6. Compute the distance from each cluster center *c*, to the document collection center *d* as *1-cos(d,c)*.

Collection Center

1. thomas	11. nomination	21. bush
2. judge	12. rights	22. people
3. court	13. marshall	23. man
4. senate	14. law	24. charges
5. supreme	15. confirmation	25. conservative
6. clarence	16. case	26. life
7. justice	17. thomas's	27. harassment
8. hearings	18. president	28. washington
9. black	19. nominee	29. vote
10. committee	20. views	30. sexual

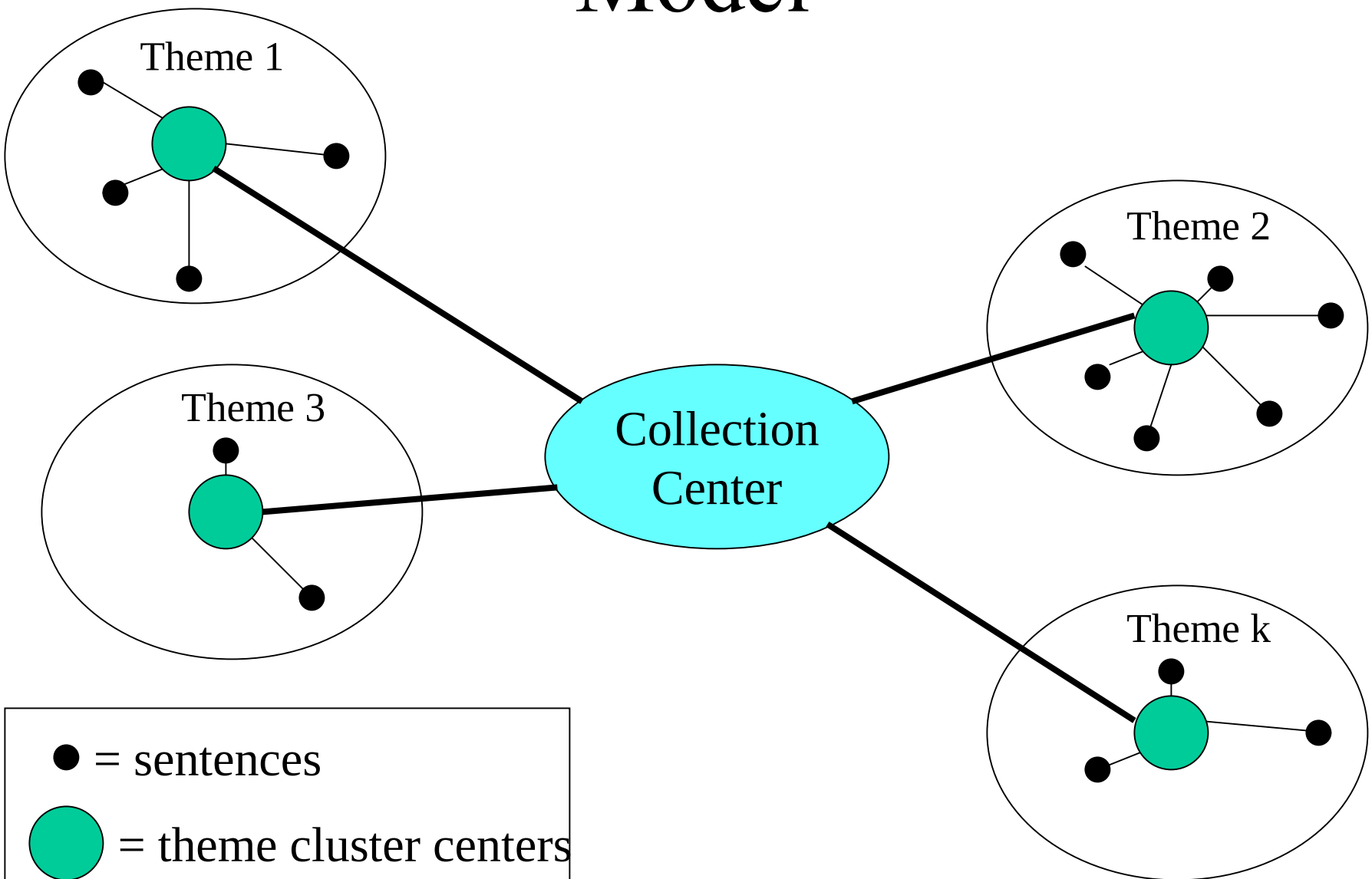
Model


Theme 1
Theme 2
Theme 3
Collection Center
Theme k
= sentences
= theme cluster centers

Model

1. Parse the document collection into a set of sentences.
2. Utilize classic (unsupervised) cluster analysis techniques to partition the set of sentences into theme clusters, i.e. disjoint subsets of sentences, such that each sentence in a cluster is "about" the same theme.
3. Compute the cluster centers as $(\mu_1, \ldots, \mu_n)$, where μ_i is the average frequency of the i^{th} term.
4. For each cluster, compute the distance from each sentence *s*, to its cluster center *c*, as *1-cos(s,c)*.
5. Consider the document collection center (modeling what the entire collection is "about") to be the term frequency vector of the entire collection.
6. Compute the distance from each cluster center *c*, to the document collection center *d* as *1-cos(d,c)*.

Model


Theme 1
Theme 2
Theme 3
Collection Center
Theme k
= sentences
= theme cluster centers

Summary Creation

1. Sentences within each cluster, are ranked in increasing order of their distance from the cluster center. Represents similarity of sentences and the theme which the cluster is “about”.
2. Clusters are ordered by their distance to the document collection. Represent how “important” it is for a theme to be included in the summary.
3. One sentence is repeatedly selected from each cluster and added to the summary, until the length constraint is violated.

Summary Creation

Theme 1
Sentence 3
Sentence 25
Sentence 7
Sentence 16

Theme 2
Sentence 45
Sentence 2
Sentence 78
Sentence 11
Sentence 18
Sentence 95

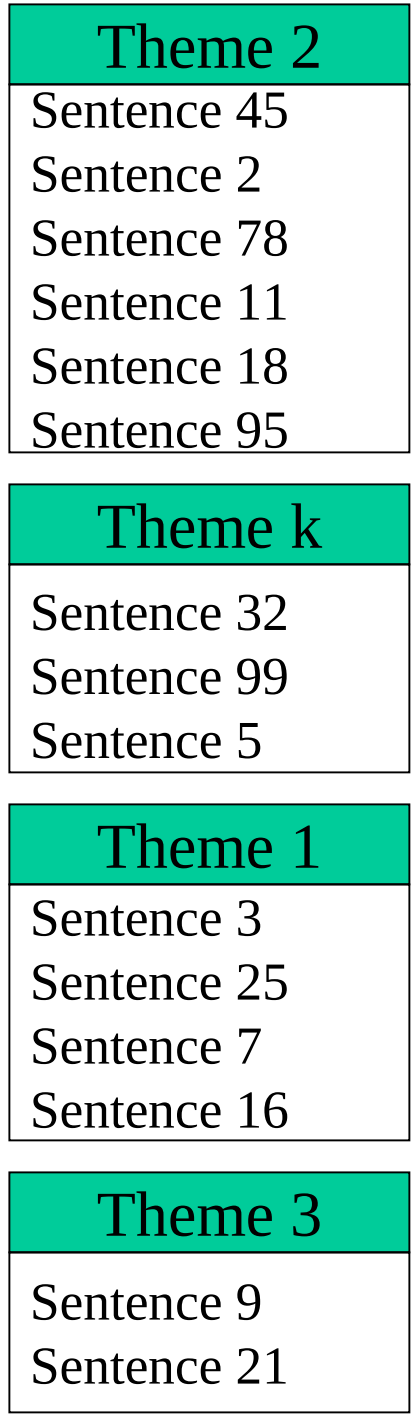
Theme 3
Sentence 9
Sentence 21

Theme k
Sentence 32
Sentence 99
Sentence 5

Summary Creation

1. Sentences within each cluster, are ranked in increasing order of their distance from the cluster center. Represents similarity of sentences and the theme which the cluster is “about”.
2. Clusters are ordered by their distance to the document collection center. Represent how “important” it is for a theme to be included in the summary.
3. One sentence is repeatedly selected from each cluster and added to the summary, until the length constraint is violated.

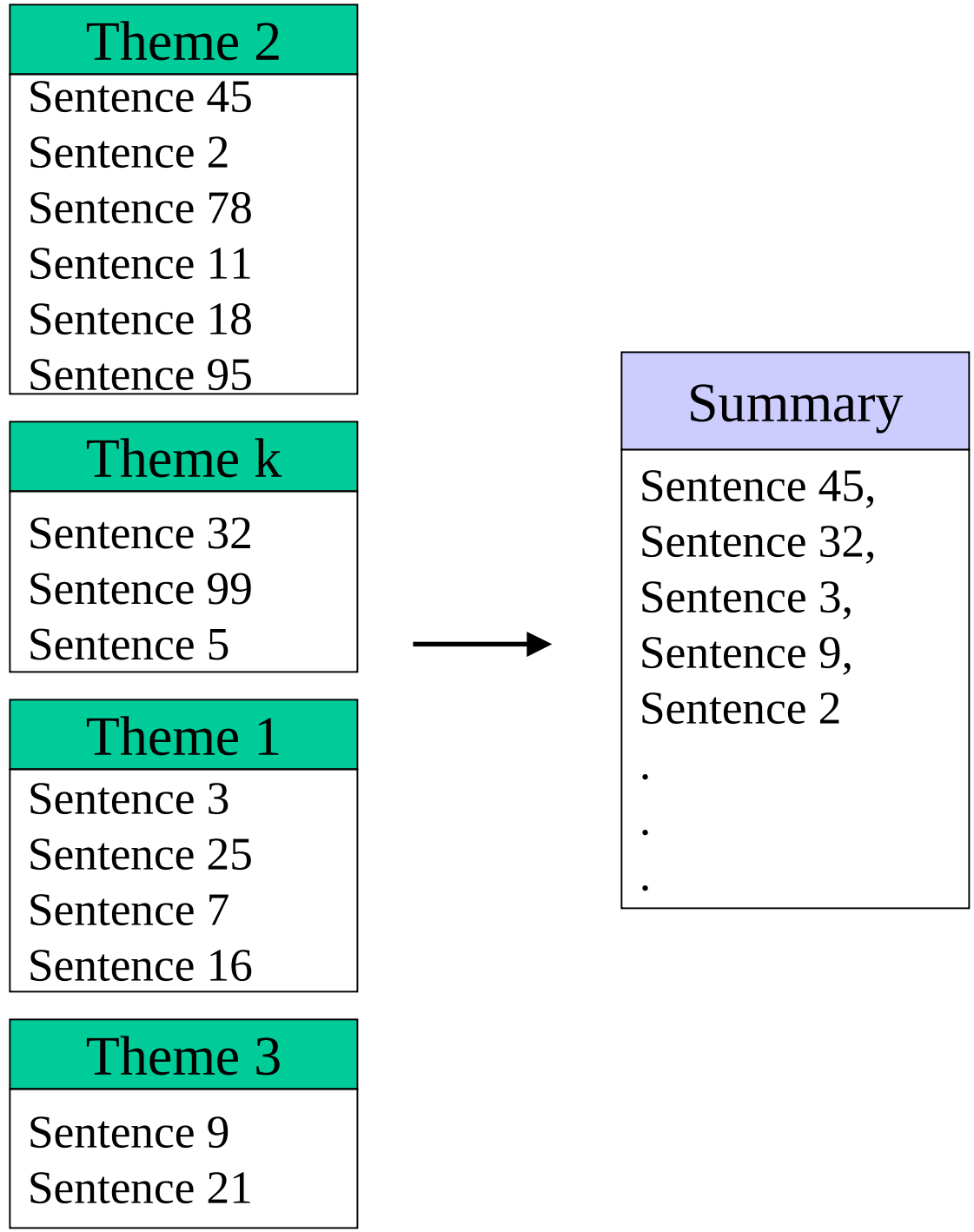
Summary Creation



Summary Creation

1. Sentences within each cluster, are ranked in increasing order of their distance from the cluster center. Represents similarity of sentences and the theme which the cluster is “about”.
2. Clusters are ordered by their distance to the document collection center. Represent how “important” it is for a theme to be included in the summary.
3. Using a “round-robin” approach, one sentence is repeatedly selected from each cluster and added to the summary, until the length constraint is violated.

Summary Creation



Details and Refinements

- Terms must appear in more than 2% of sentences to be used in clustering.
- Only sentences having between 5 and 30 words are used.

20-20 Hindsight

- Clustering of sentences did not yield intuitively encouraging results. Why?
- Need to do a better job of “noise” removal, i.e. feature selection (e.g. in the collection center) only the most important terms should have been used. Might help in ordering the clusters.

Example 50 Word Summary

If the effort against Judge Thomas fizzles and the organizations are seen to have grossly overreached, their credibility will be badly damaged in any future battles. Can Clarence Thomas be confirmed to the Supreme Court? Studied for priesthood; Thomas attended Roman Catholic schools through college and studied for the priesthood.

Evaluation

- The data for each topic/system/size (T/S/S) combination end with a set of triples (*x*,*y*,*z)* of which *y* seems to represent the degree to which the unit, for this triple, “covers” the model concept to which it is most closely aligned.
- For each T/S/S we compute the average <*y*> over all triples

Evaluation (2)

- For each T/S/S we then sort the average value, and assign to each T/S/S the rank of that average (1=highest value).
- The performance of a system, for a given size of summary, is the average, over T, of the rank. Call this <r>/S/S.
- Our average rank is about 23 or so. Discouraging

Evaluation (3)

- There are some difficulties:
  - the number of cases for each -/S/S is not the same for all topics
- This can be used to establish confidence levels for differences between systems
  - it is a Friedman type test.

Histogram of the ranks we achieved (Multi-doc): U all sizes together

- 3 ******
- 4 *
- 5 *****
- 6 ***
- 7 ******
- 8 *******
- 9 *********
- 10 *******
- 11 *****
- 12 *****
- 13 ********
- 14 *******
- 15 **********
- 16 ******************
- 17 ********************

Thank you!