

# Interesting-Phrase Mining for Ad-Hoc Text Analytics

Klaus Berberich  
([kberberi@mpi-inf.mpg.de](mailto:kberberi@mpi-inf.mpg.de))

**Joint work with:**  
Srikanta Bedathur, Jens Dittrich,  
Nikos Mamoulis, and Gerhard Weikum  
(originally presented at VLDB 2010)



# Motivation

- Continuously growing wealth of **unstructured text data**, e.g.:
  - **e-mail** and **instant messaging**
  - **social media content** (e.g., twitter and facebook)
  - **customer reports** and **product reviews**
  - **general Web** (e.g., news portals)
- **Search** on this data is understood and **under control** (e.g., find all news articles that mention **barack obama**)
- **Gaining insights is tedious** and a task mostly left to users (e.g., identify characteristic quotations by **barack obama**)



# Motivation

- **Tools** available today include
  - Tag clouds
  - Search-result snippets
- **Limitations:**
  - **focus on single terms** – no entity names, quotations, etc.
  - **frequency-based** – not necessarily interesting (e.g., stopwords)
  - **global analysis** – no ad-hoc document sets (e.g., determined by query)
- **Our idea:** Identify **interesting phrases** that distinguish an **ad-hoc document set** from the document collection as a whole







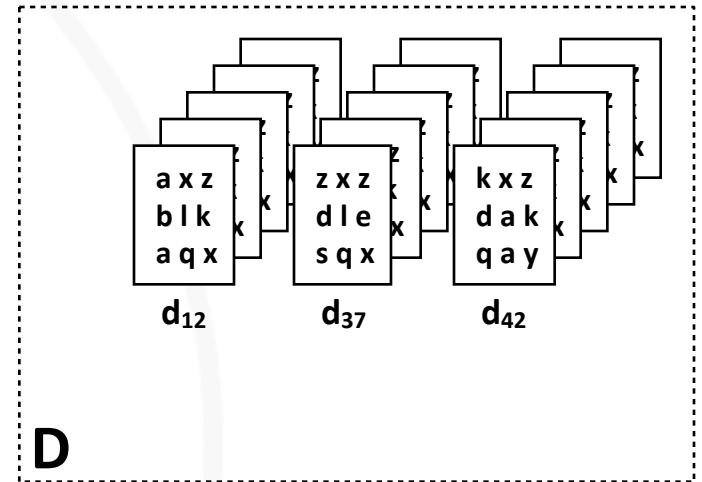
# Outline

- Motivation
- Problem Statement
- Our Approach
- Prior Art
- Experimental Evaluation
- Conclusion & Ongoing Research



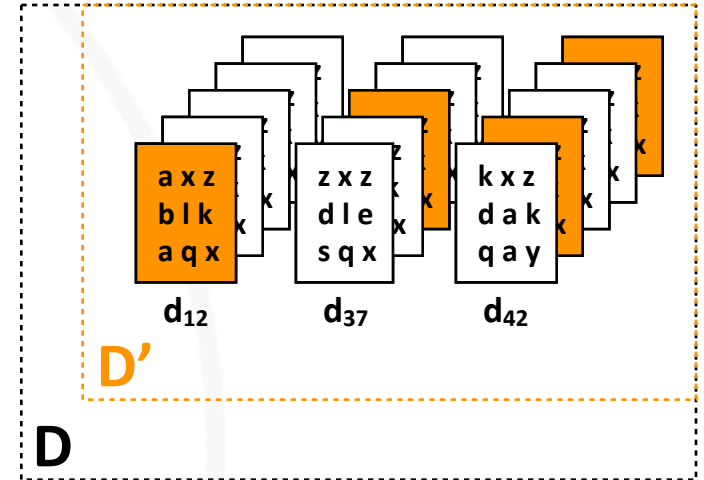
# Model & Problem Statement

- **Document collection  $D$**
- **Documents** are sequences of terms (e.g.,  $d_{12} = \langle a x z b l k a q x \rangle$ )
- **Phrases** are sequences of terms (e.g.,  $\langle a x z \rangle$ ,  $\langle a x z b \rangle$ ,  $\langle z b l k \rangle$ , ...)
- **Input: Ad-hoc document set  $D' \subseteq D$**  (e.g., all documents that contain **barack obama**)
- **Output:  $k$  most interesting phrases in  $D'$**



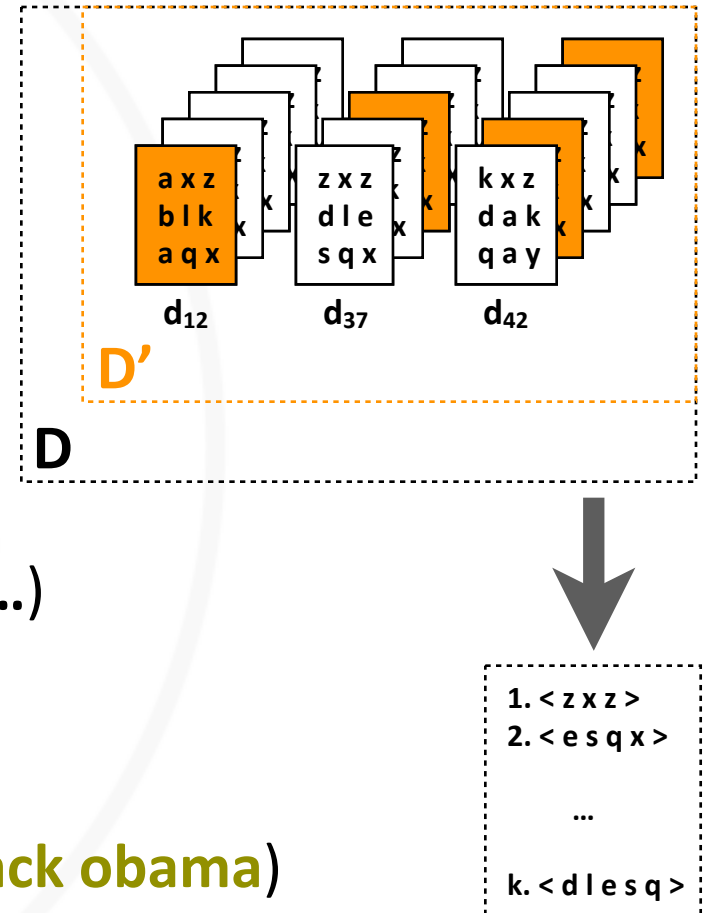
# Model & Problem Statement

- **Document collection  $D$**
- **Documents** are sequences of terms (e.g.,  $d_{12} = \langle a x z b l k a q x \rangle$ )
- **Phrases** are sequences of terms (e.g.,  $\langle a x z \rangle$ ,  $\langle a x z b \rangle$ ,  $\langle z b l k \rangle$ , ...)
- **Input: Ad-hoc document set  $D' \subseteq D$**  (e.g., all documents that contain **barack obama**)
- **Output:  $k$  most interesting phrases in  $D'$**



# Model & Problem Statement

- Document collection  $D$
- Documents are sequences of terms (e.g.,  $d_{12} = \langle a x z b l k a q x \rangle$ )
- Phrases are sequences of terms (e.g.,  $\langle a x z \rangle$ ,  $\langle a x z b \rangle$ ,  $\langle z b l k \rangle$ , ...)
- Input: Ad-hoc document set  $D' \subseteq D$  (e.g., all documents that contain **barack obama**)
- Output:  $k$  most interesting phrases in  $D'$



# When Do We Consider a Phrase Interesting?

- We distinguish for a phrase  $p$  its
  - **local frequency**  $\text{freq}(p, D')$  in the ad-hoc document set  $D'$
  - **global frequency**  $\text{freq}(p, D)$  in the document collection  $D$
- The **interestingness** of phrase  $p$  is defined as

$$I(p, D') = \frac{\text{freq}(p, D')}{\text{freq}(p, D)}$$

- We consider only phrases as **relevant** that
  - **are globally frequent**, i.e., occur in at least  $\tau$  documents from  $D$
  - **have at most length  $m$**  (e.g., 5 in our experiments)
- **Note:** Our methods adapt to **other definitions of interestingness** (e.g., based on log-likelihood ratios or PMI)



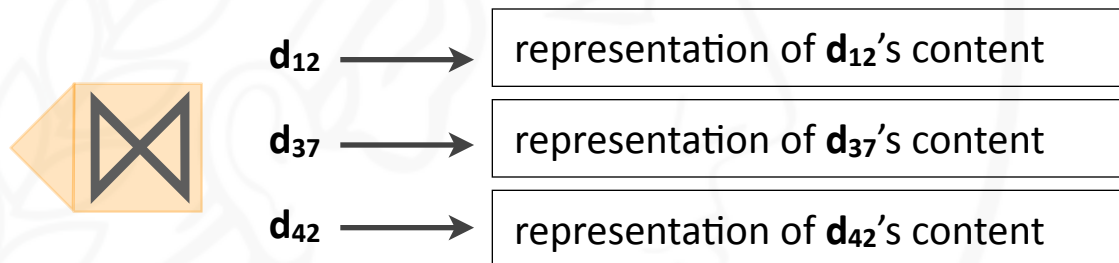
# Outline

- Motivation
- Problem Statement
- Our Approach
- Prior Art
- Experimental Evaluation
- Conclusion & Ongoing Research



# Overview

- We precompute the global frequency  $\text{freq}(\mathbf{p}, \mathbf{D})$  for any relevant phrase  $\mathbf{p}$  using an **apriori-style algorithm** and keep a **phrase dictionary** that maps  $\mathbf{p} \rightarrow \text{freq}(\mathbf{p}, \mathbf{D})$
- Our methods rely on **forward indexes** that map each document to a **representation of its content**



- For a given ad-hoc document set  $\mathbf{D}'$  our methods
  - **access** the forward index for each  $\mathbf{d} \in \mathbf{D}'$
  - **merge** the  $|\mathbf{D}'|$  content representations
  - **output** the  $k$  most interesting phrases

# Document Content

- **Idea:** Keep **document content explicitly** as a sequence of contained terms (or, term identifiers)



- **Benefit:**

- **Space-efficient**

- **Drawbacks:**

- **Enumeration** of phrases needed (including globally-infrequent ones)
- **Requires phrase dictionary** with global frequencies  $\text{freq}(p, D)$

# Phrases

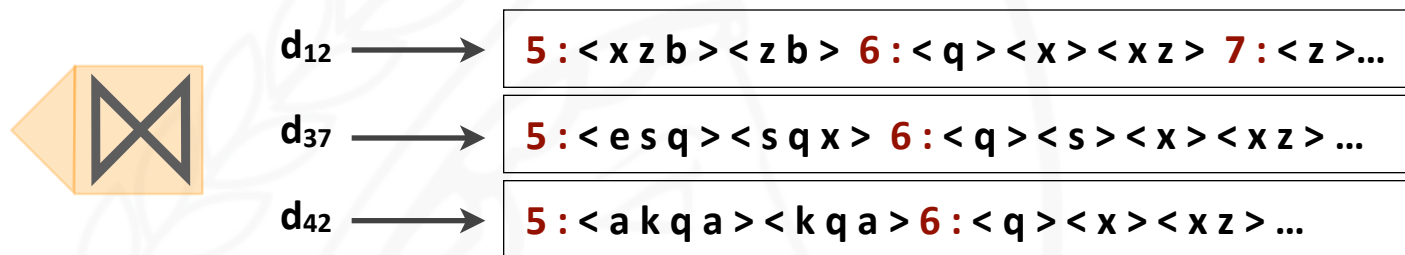
- **Idea:** Keep **globally-frequent phrases** contained in document in a consistent (e.g., lexicographical) order



- **Benefits:**
  - Considers **only globally-frequent phrases**
  - Consistent sort order **facilitates merging**
- **Drawbacks:**
  - **Space-inefficient**
  - **Requires phrase dictionary** with global frequencies  $\text{freq}(p, D)$

# Frequency-Ordered Phrases

- **Idea:** Keep **contained globally-frequent phrases** in ascending order of their embedded global frequencies **freq(p, D)**



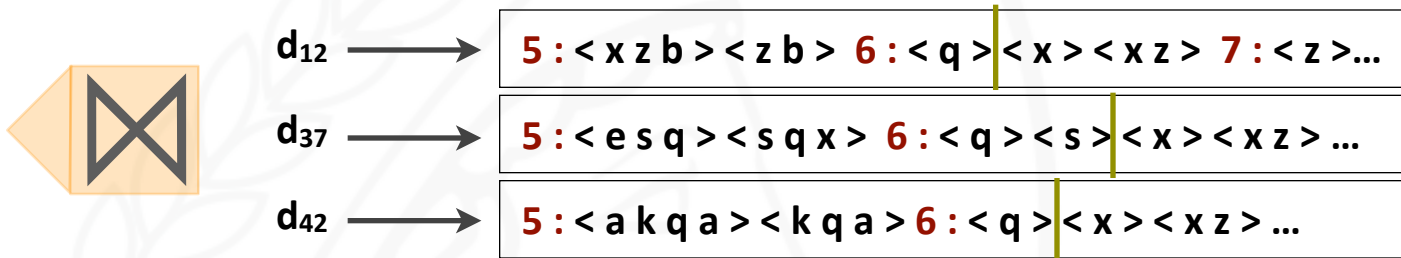
- **Interestingness** of any unseen phrase is **upper-bounded** by

$$\min \left\{ 1, \frac{|D'|}{\text{freq}(p, D)} \right\}$$

where  $p$  is the last phrase encountered

# Frequency-Ordered Phrases

- **Idea:** Keep **contained globally-frequent phrases** in ascending order of their embedded global frequencies  $\text{freq}(\mathbf{p}, \mathbf{D})$



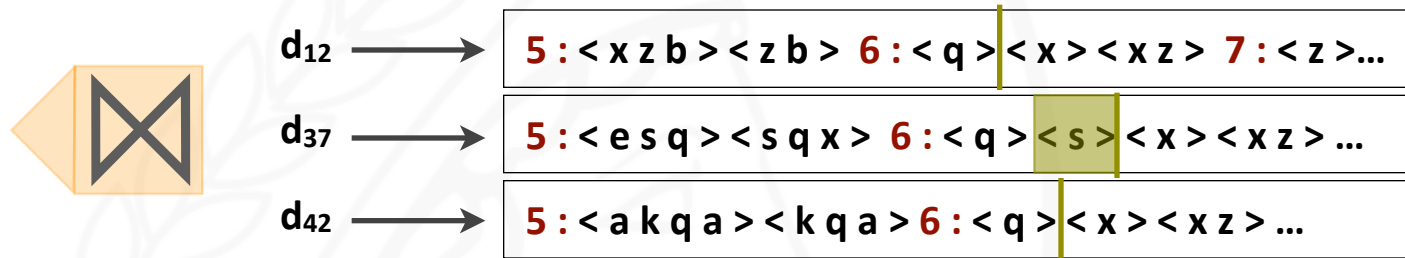
- **Interestingness** of any unseen phrase is **upper-bounded** by

$$\min \left\{ 1, \frac{|\mathbf{D}'|}{\text{freq}(\mathbf{p}, \mathbf{D})} \right\}$$

where  $\mathbf{p}$  is the last phrase encountered

# Frequency-Ordered Phrases

- **Idea:** Keep **contained globally-frequent phrases** in ascending order of their embedded global frequencies **freq(p, D)**



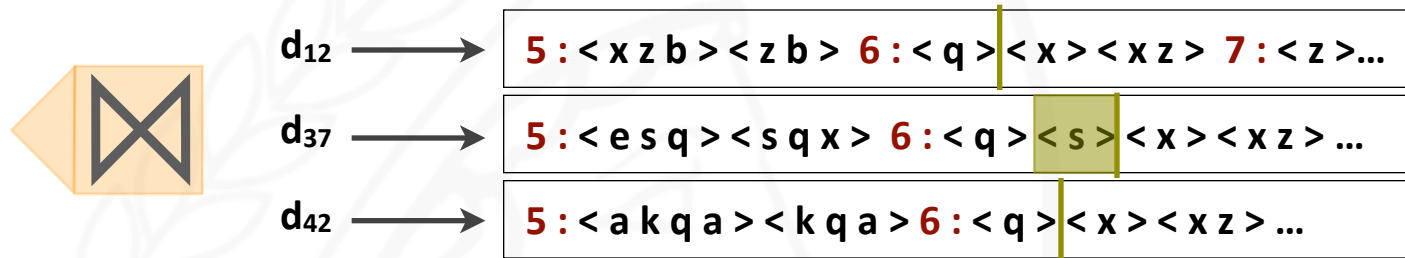
- **Interestingness** of any unseen phrase is **upper-bounded** by

$$\min \left\{ 1, \frac{|D'|}{\text{freq}(p, D)} \right\}$$

where **p** is the last phrase encountered

# Frequency-Ordered Phrases

- **Idea:** Keep **contained globally-frequent phrases** in ascending order of their embedded global frequencies **freq(p, D)**



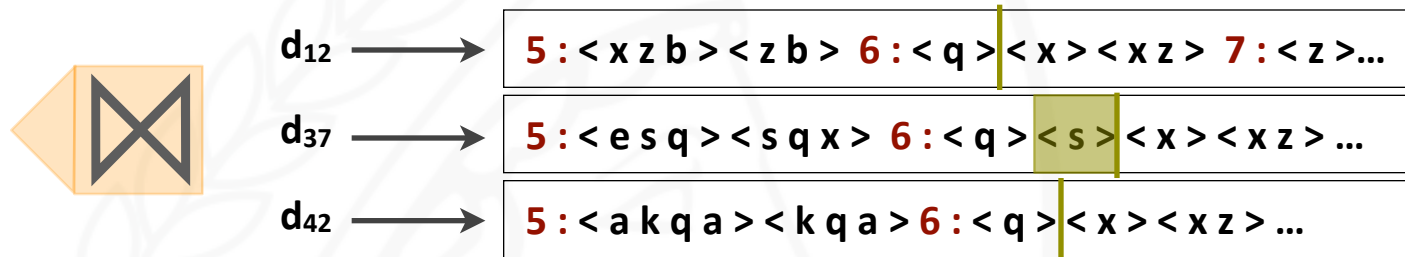
- **Interestingness** of any unseen phrase is **upper-bounded** by

$$\min \left\{ 1, \frac{|D'|}{\text{freq}(p, D)} \right\} \frac{|D'|}{\text{freq}(\langle s \rangle, D)} = \frac{3}{6}$$

where **p** is the last phrase encountered

# Frequency-Ordered Phrases

- **Idea:** Keep **contained globally-frequent phrases** in ascending order of their embedded global frequencies **freq(p, D)**



- **Benefits:**

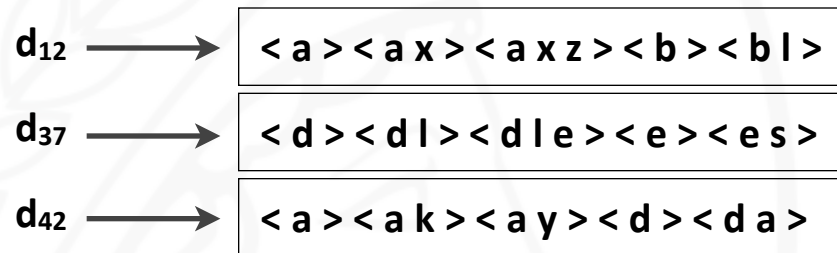
- **Early termination** possible when no unseen phrase can make it into the top-**k** of most interesting phrases
- **Self-contained** (i.e., no phrase dictionary needed)

- **Drawback:**

- **Space-inefficient**

# Prefix-Maximal Phrases

- **Observation:** Globally-frequent phrases are often redundant and therefore do not need to be kept explicitly



- **Definition:** A phrase  $p$  is prefix-maximal in document  $d$  if
  - $p$  is globally-frequent
  - $d$  does not contain a globally-frequent phrase  $p'$  of which  $p$  is a prefix
- A prefix-maximal phrase (e.g.,  $\langle axz \rangle$ ) can represent all its prefixes (i.e.,  $\langle a \rangle$  and  $\langle ax \rangle$ ) and it's guaranteed that they're globally-frequent and contained in  $d$



# Prefix-Maximal Phrases

- **Observation:** Globally-frequent phrases are **often redundant** and therefore **do not need to be kept explicitly**

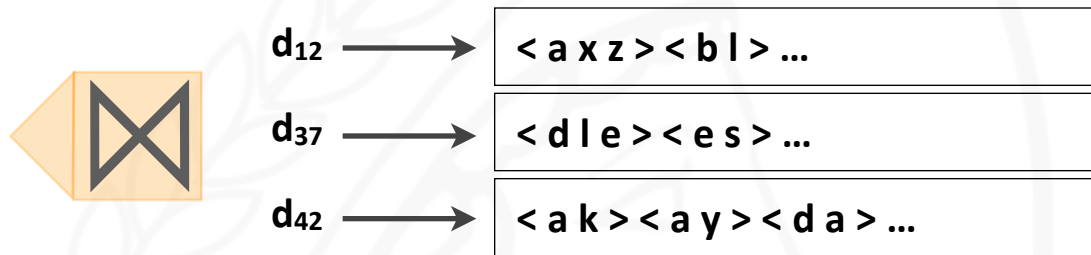


- **Definition:** A phrase **p** is **prefix-maximal in document d** if
  - **p** is **globally-frequent**
  - **d** does not contain a **globally-frequent phrase p'** of which **p** is a **prefix**
- A prefix-maximal phrase (e.g., < a x z >) can **represent all its prefixes** (i.e., < a > and < a x >) and it's guaranteed that they're globally-frequent and contained in **d**



# Prefix-Maximal Phrases

- **Idea:** Keep **contained prefix-maximal phrases** in lexicographical order and **extract prefixes on-the-fly**



- **Benefit:**

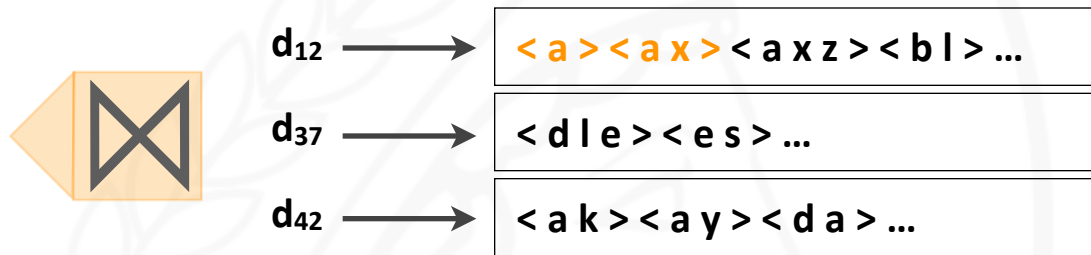
- **Space-efficient**

- **Drawbacks:**

- **Extraction of prefixes** entails additional bookkeeping
- **Requires phrase dictionary** with global frequencies  $\text{freq}(p, D)$

# Prefix-Maximal Phrases

- **Idea:** Keep **contained prefix-maximal phrases** in lexicographical order and **extract prefixes on-the-fly**



- **Benefit:**

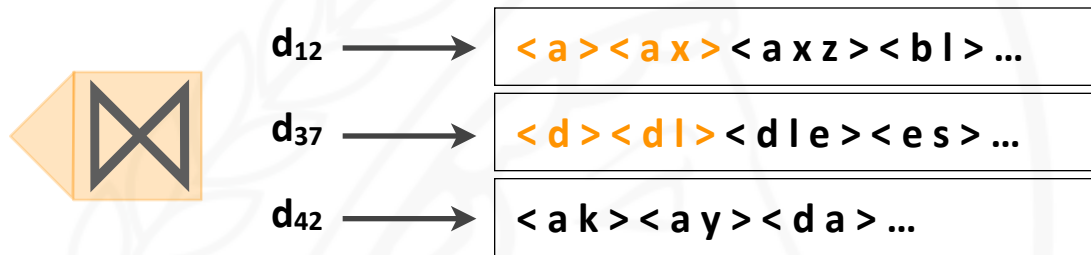
- **Space-efficient**

- **Drawbacks:**

- **Extraction of prefixes** entails additional bookkeeping
- **Requires phrase dictionary** with global frequencies  $\text{freq}(p, D)$

# Prefix-Maximal Phrases

- **Idea:** Keep **contained prefix-maximal phrases** in lexicographical order and **extract prefixes on-the-fly**



- **Benefit:**

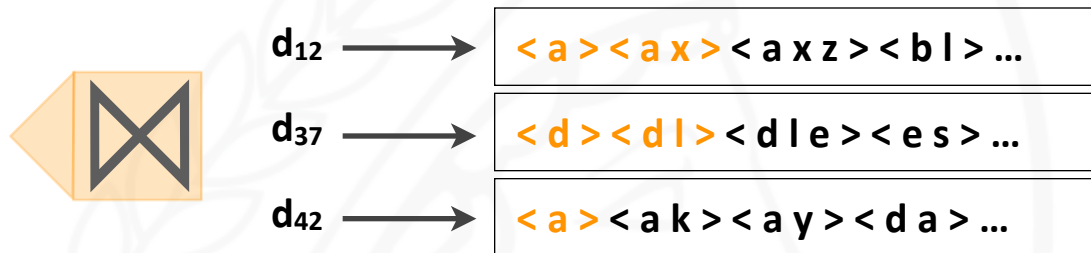
- **Space-efficient**

- **Drawbacks:**

- **Extraction of prefixes** entails additional bookkeeping
- **Requires phrase dictionary** with global frequencies  $\text{freq}(\mathbf{p}, \mathbf{D})$

# Prefix-Maximal Phrases

- **Idea:** Keep **contained prefix-maximal phrases** in lexicographical order and **extract prefixes on-the-fly**



- **Benefit:**

- **Space-efficient**

- **Drawbacks:**

- **Extraction of prefixes** entails additional bookkeeping
- **Requires phrase dictionary** with global frequencies  $\text{freq}(p, D)$

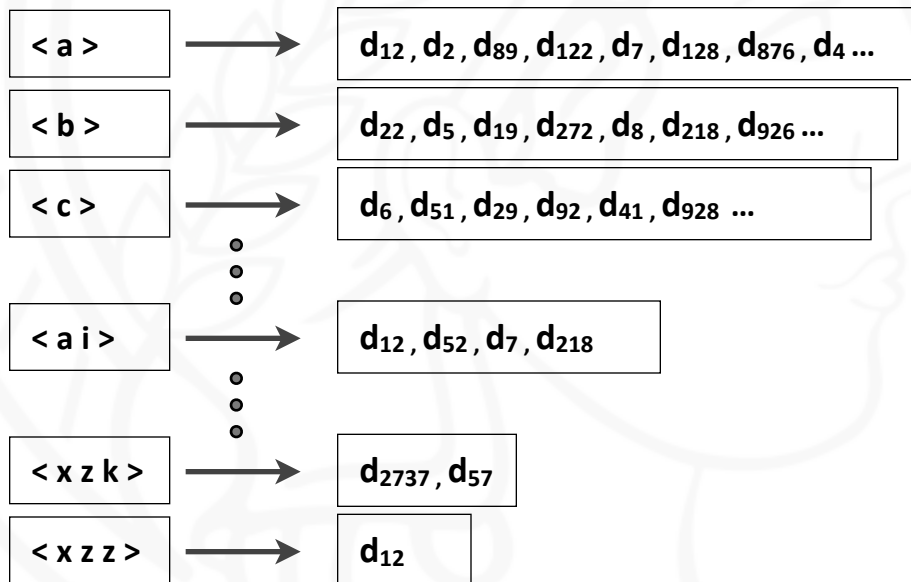
# Outline

- Motivation
- Problem Statement
- Our Approach
- Prior Art
- Experimental Evaluation
- Conclusion & Ongoing Research



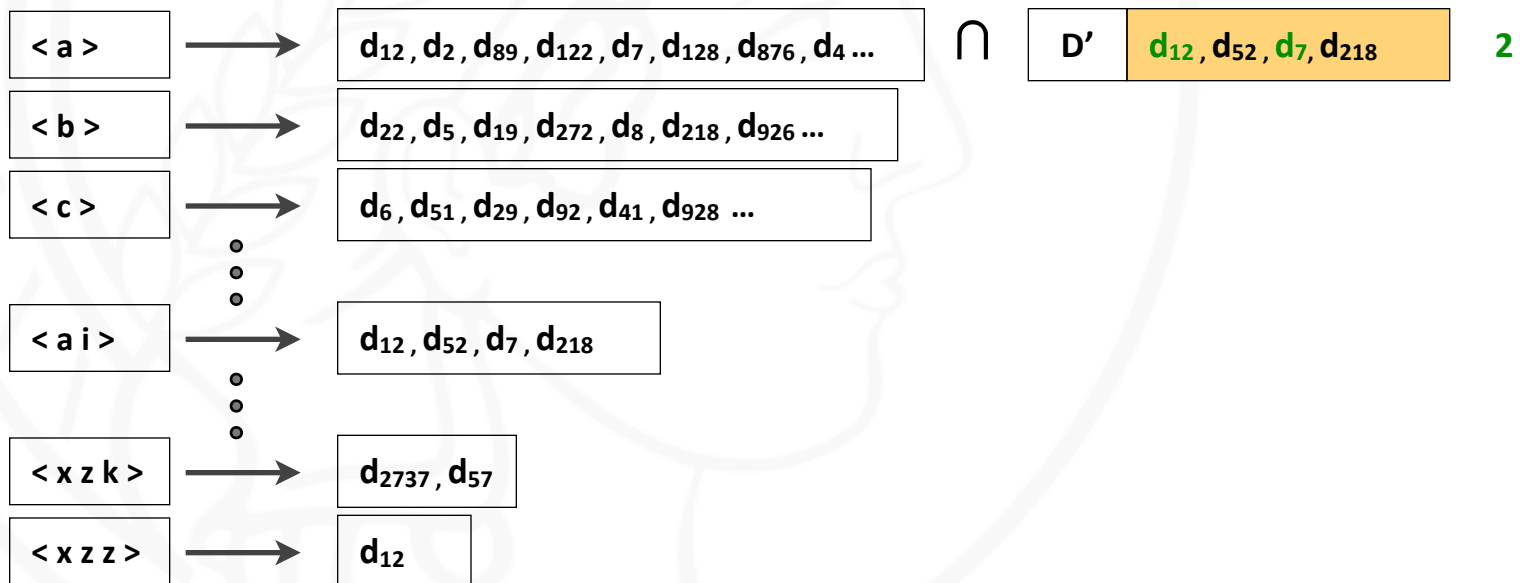
# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



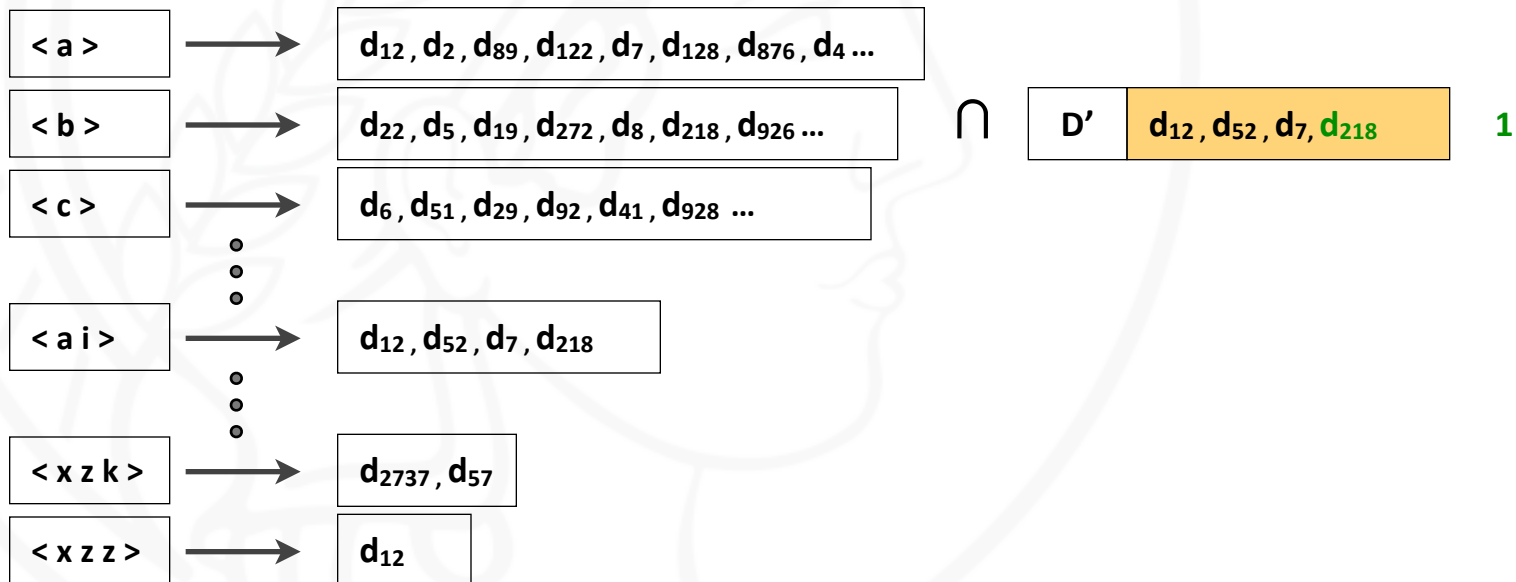
# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



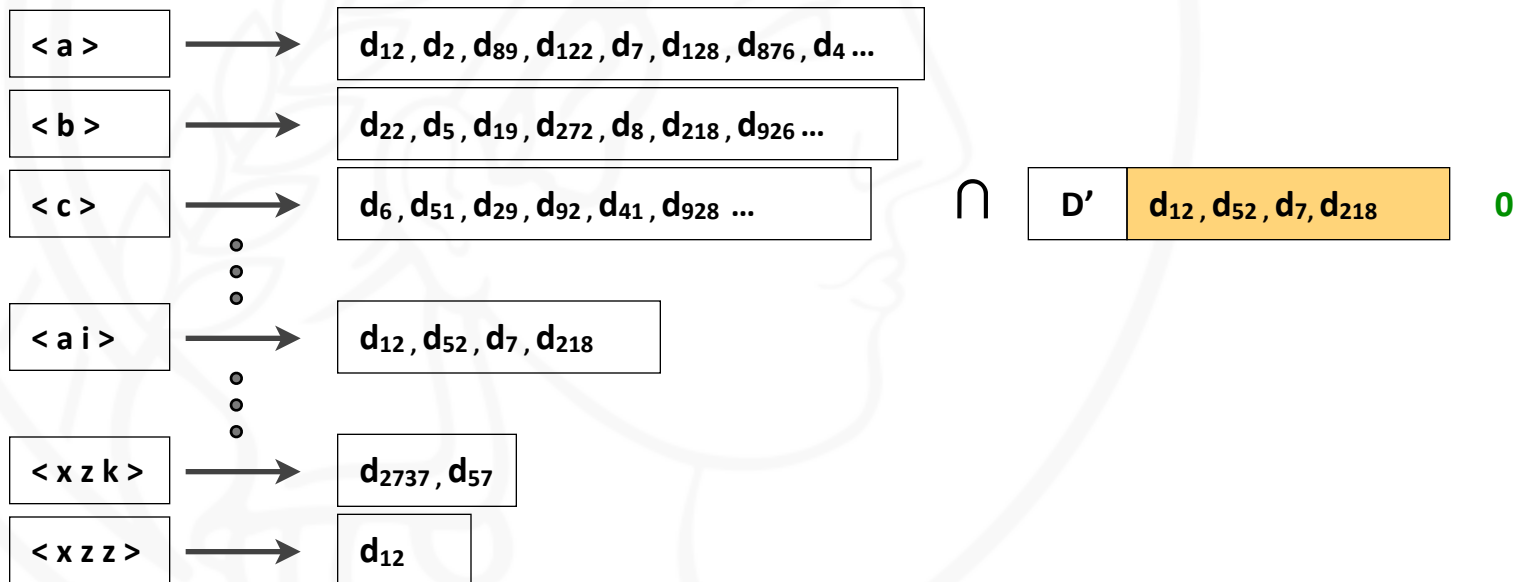
# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



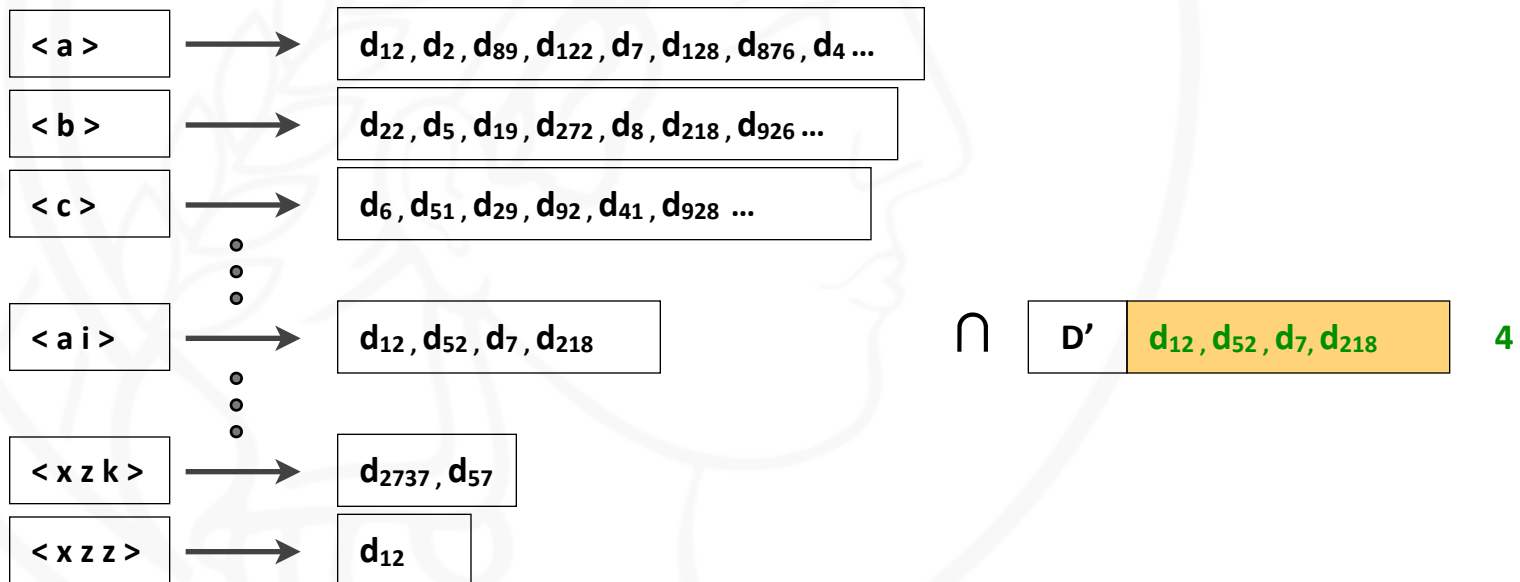
# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



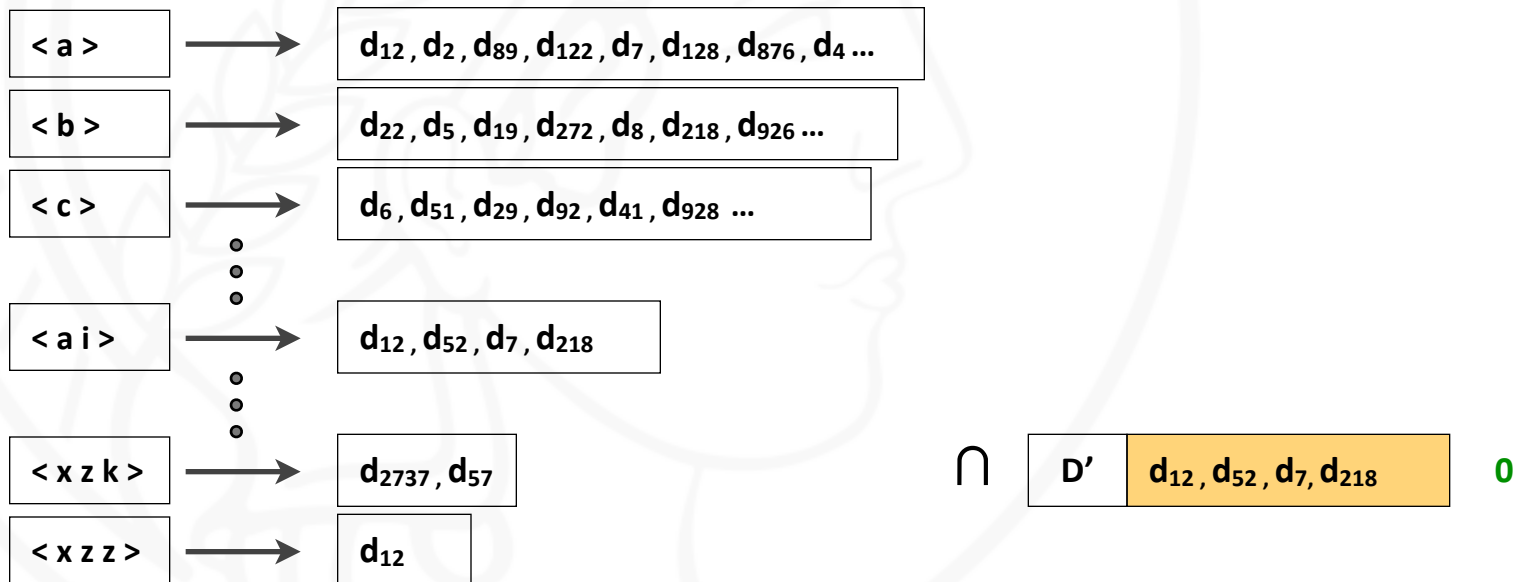
# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



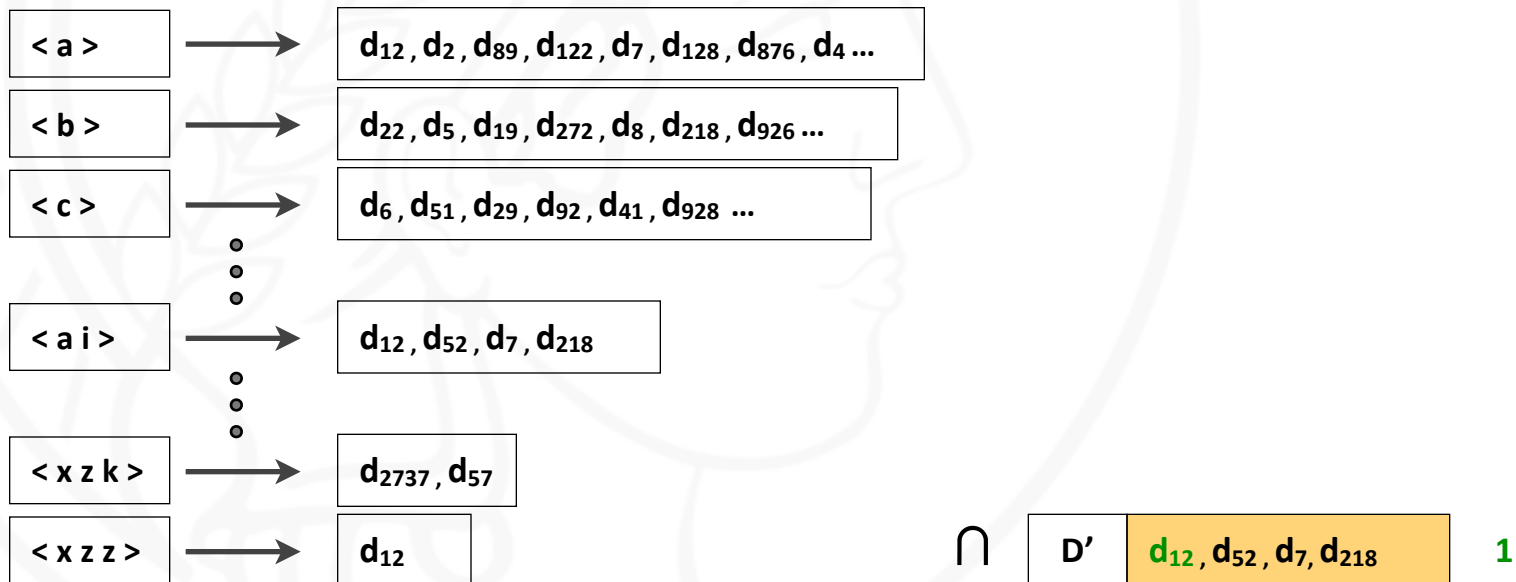
# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



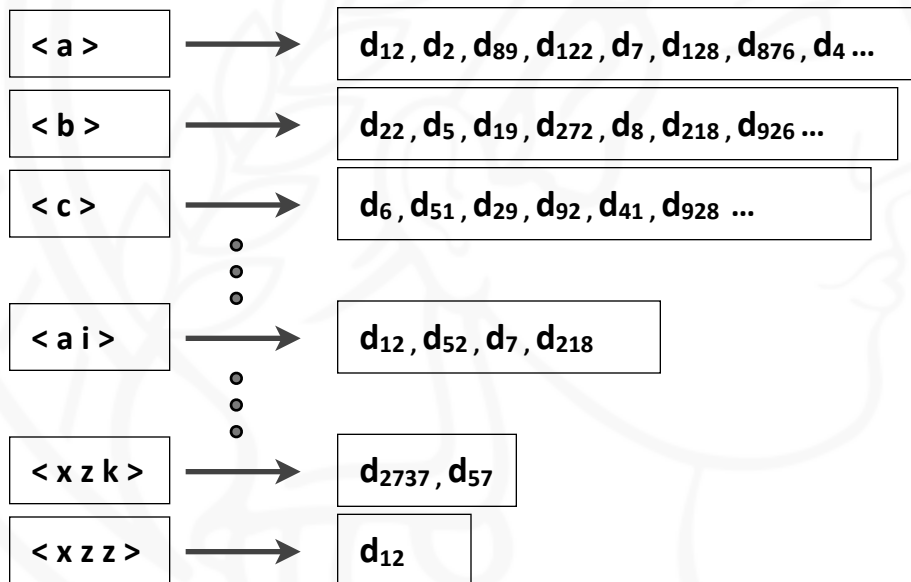
# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



# MCX: Multidimensional Content eXploration

- **Objective:** Identify **k most frequent relevant phrases** in a given ad-hoc document set **D'**
- **Idea:** Build **inverted index** that maps  $p \rightarrow \{ d \mid d \text{ contains } p \}$  and sorts phrases in **descending order of freq(p, D)**



# MCX: Multidimensional Content eXploration

- **Optimization:**
  - **Approximate fast set intersections** through order randomization
- **Benefits:**
  - **Early termination** possible when no unseen phrase can make it into the top-k of most frequent phrases
  - performs well for **very large ad-hoc document sets  $D'$**
- **Drawback:**
  - **considers only frequency** as a coarse-grained pre-filter; re-ranks identified frequent phrases based on interestingness
- **Reference:** A. Simitsis, A. Baid, Y. Sismanis, and B. Reinwald. *Multidimensional Content eXploration*. In VLDB '08



# Outline

- Motivation
- Problem Statement
- Our Approach
- Prior Art
- Experimental Evaluation
- Conclusion & Ongoing Research



# Experimental Evaluation

- **Dataset:** *The New York Times* Annotated Corpus consisting of **1.8M newspaper articles** published between 1987 and 2007
- **Queries** to determine ad-hoc document sets based on
  - **person-related** (e.g., **steve jobs, hillary clinton**,...)
  - **news-related** (e.g., **world trade center, world cup**,...)
  - **based on metadata** (e.g., **/travel/destinations/europe**,...)
- **Implementation** in Java **represents data compactly** (e.g., variable-length encoding)
- **System:** SUN server-class machine (4 CPUs / 16Gb RAM / RAID-5 / Windows 2003 Server)



# Examples of Interesting Phrases

- **Query: john lennon**

- 1) ...since john lennon was assassinated...
- 2) ...lennon's childhood...
- 3) ...post beatles work...

- **Query: bob marley**

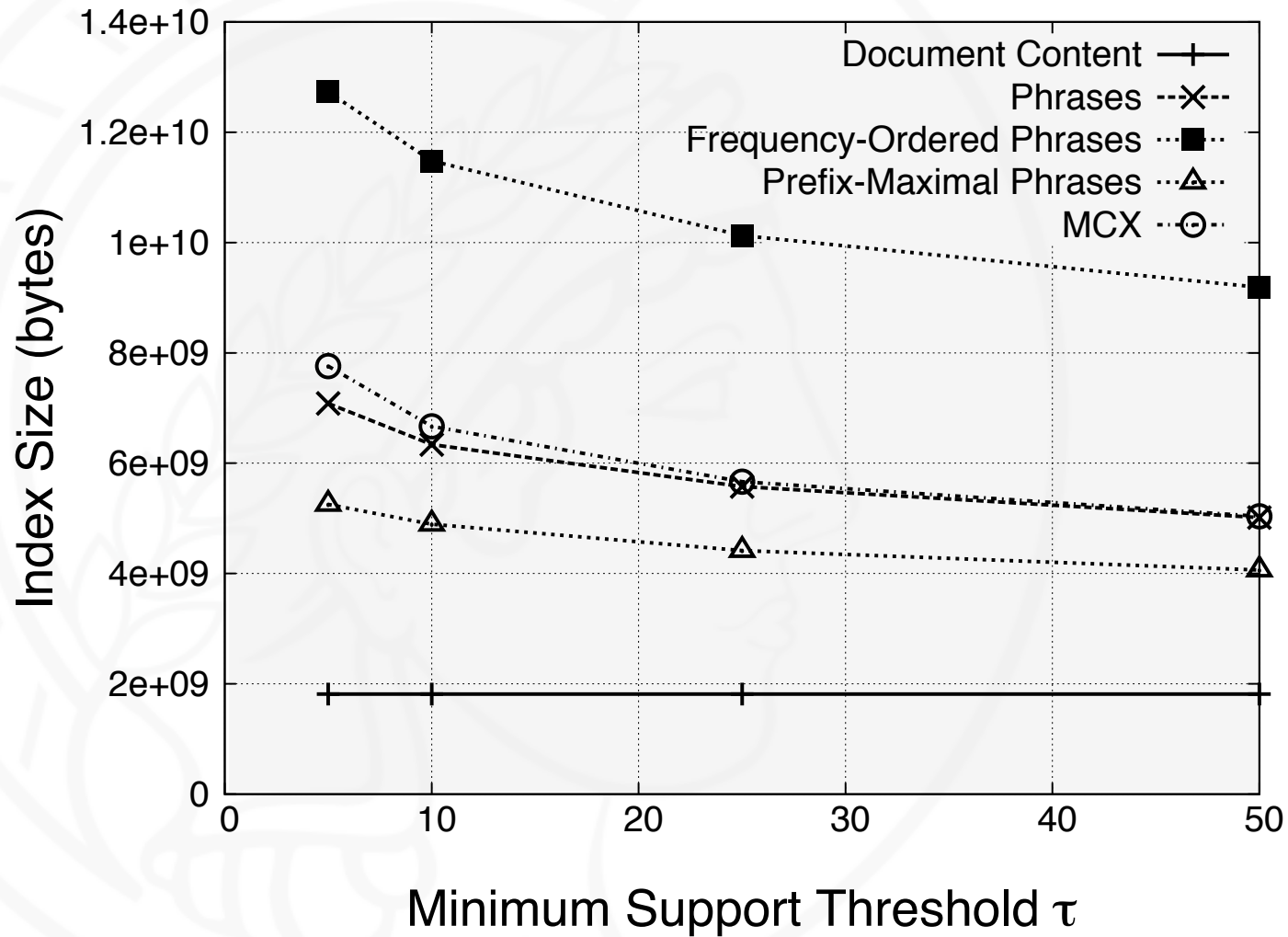
- 1) ...music of bob marley...
- 2) ...marley the jamaican musician...
- 3) ...i shot the sheriff...

- **Query: john mccain**

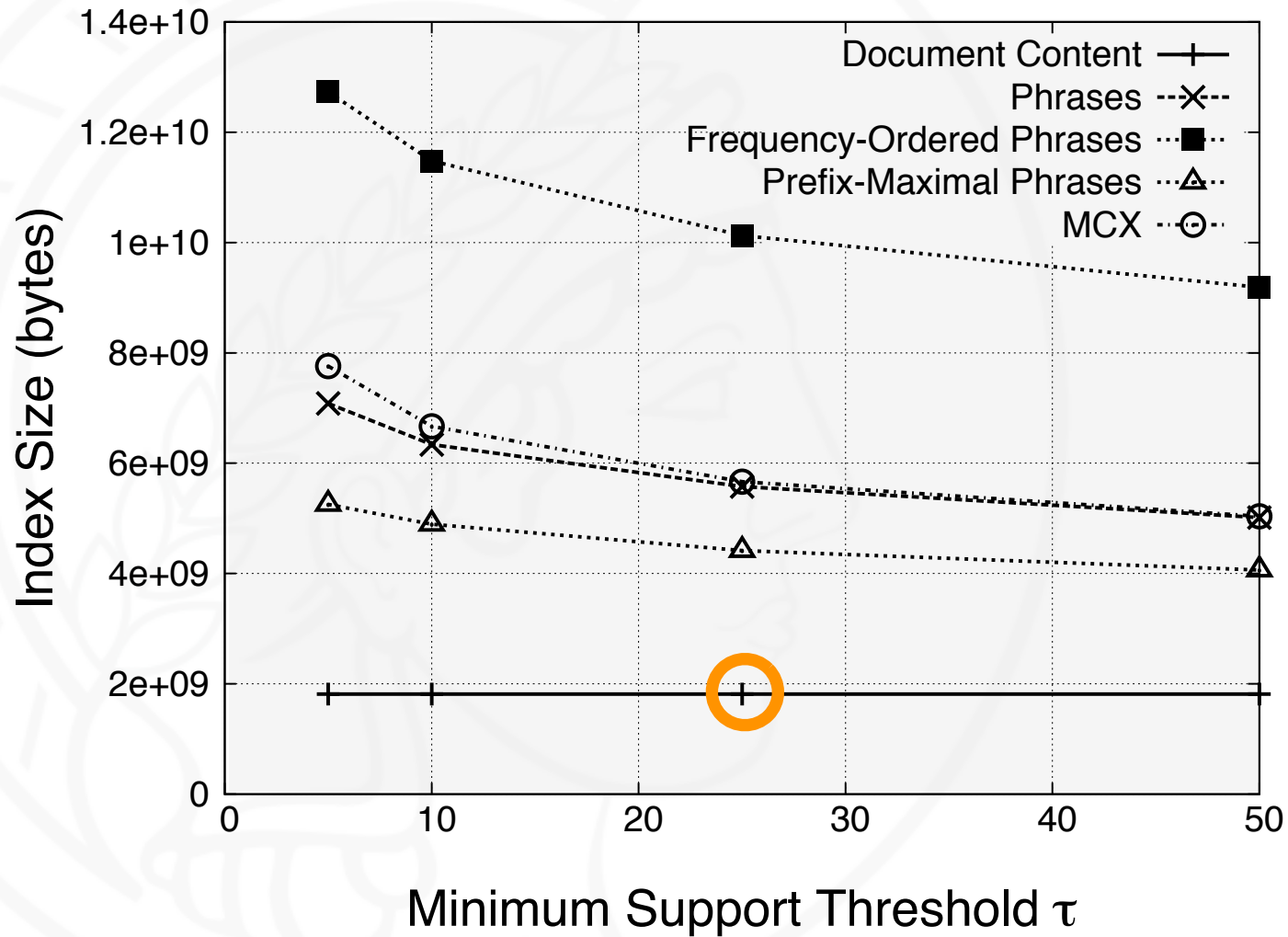
- 1) ...to beat al gore like...
- 2) ...2000 campaign in arizona...
- 3) ...the senior senator from virginia...



# Index Sizes for Different Values of $\tau$



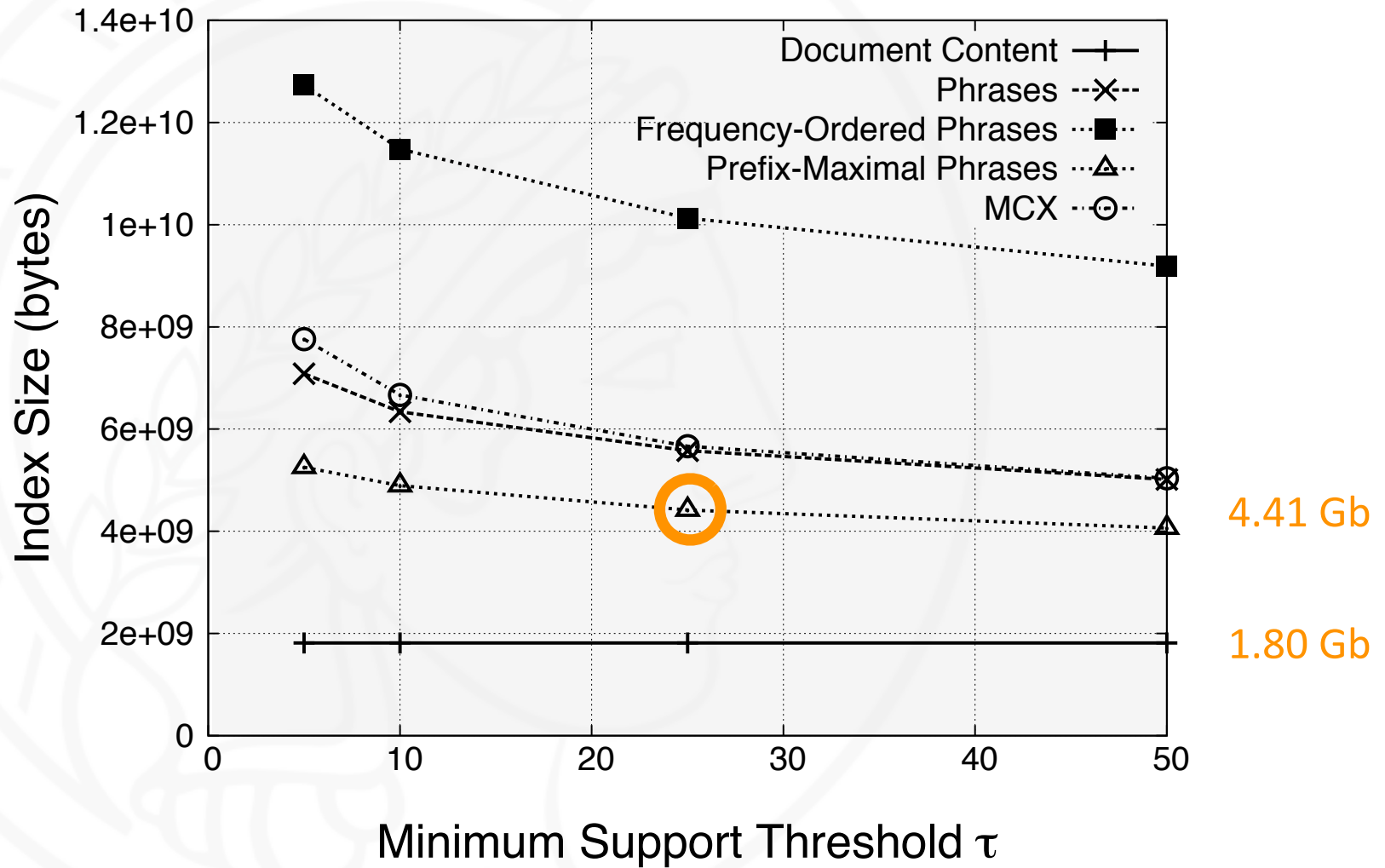
# Index Sizes for Different Values of $\tau$



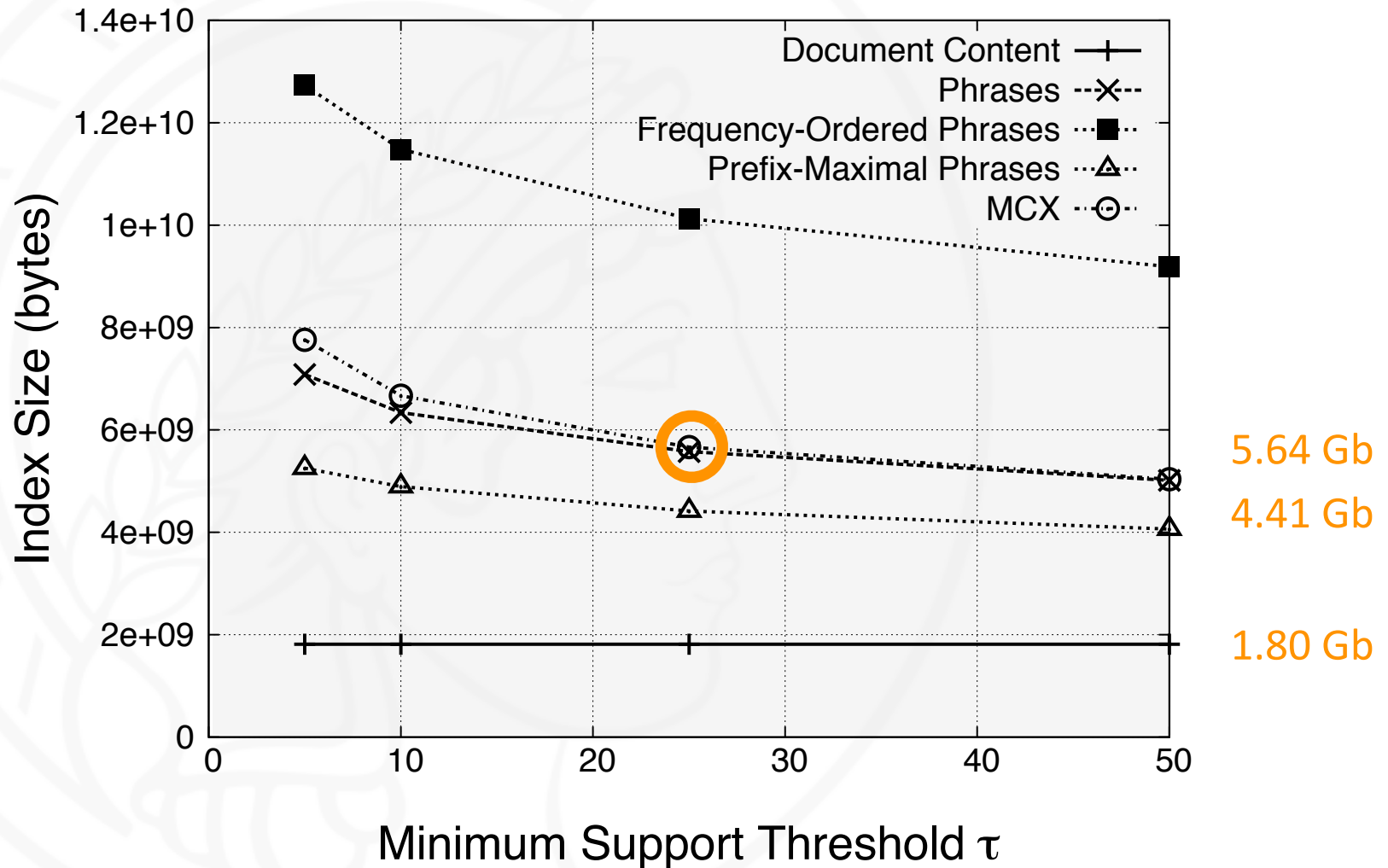
1.80 Gb



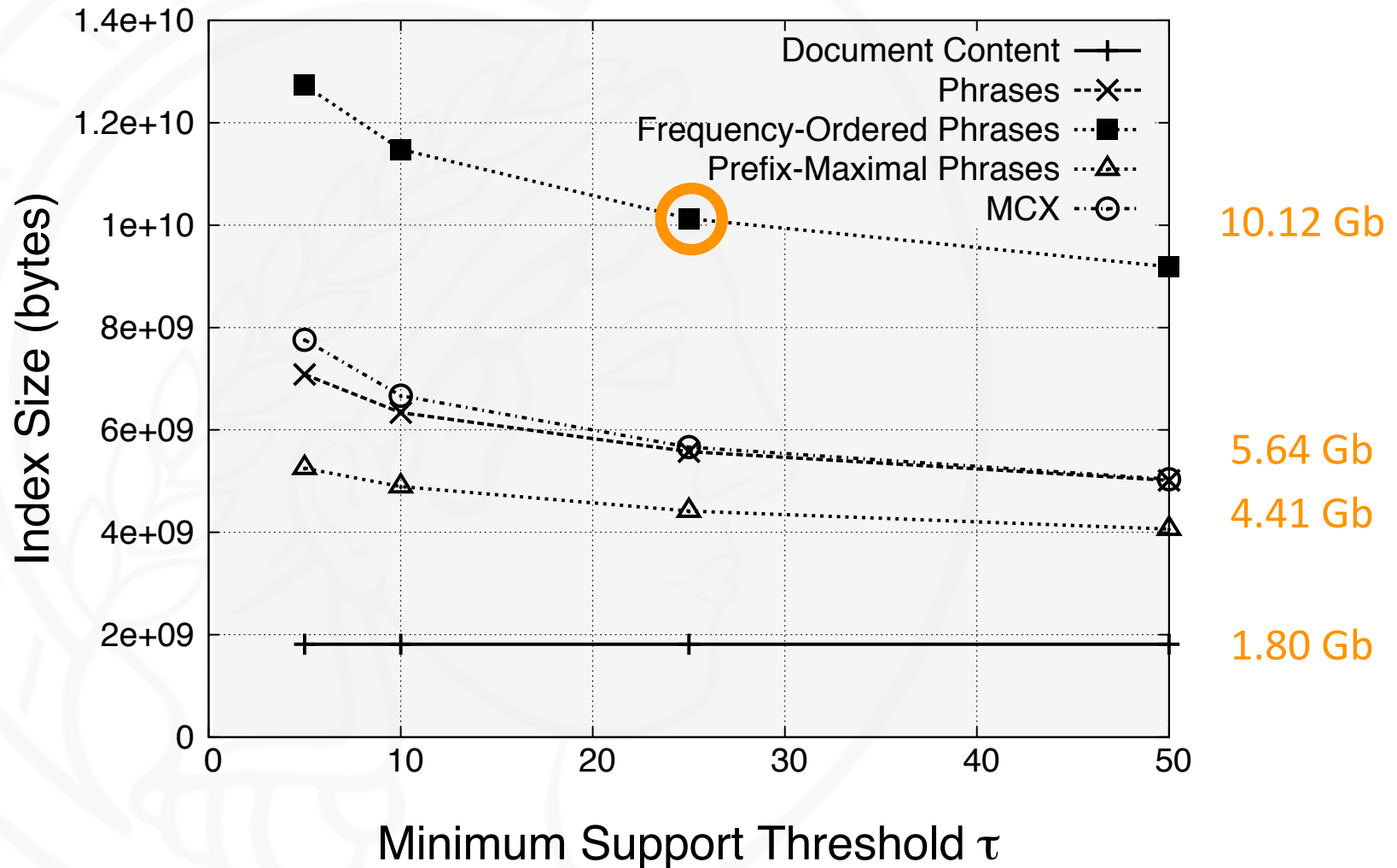
# Index Sizes for Different Values of $\tau$



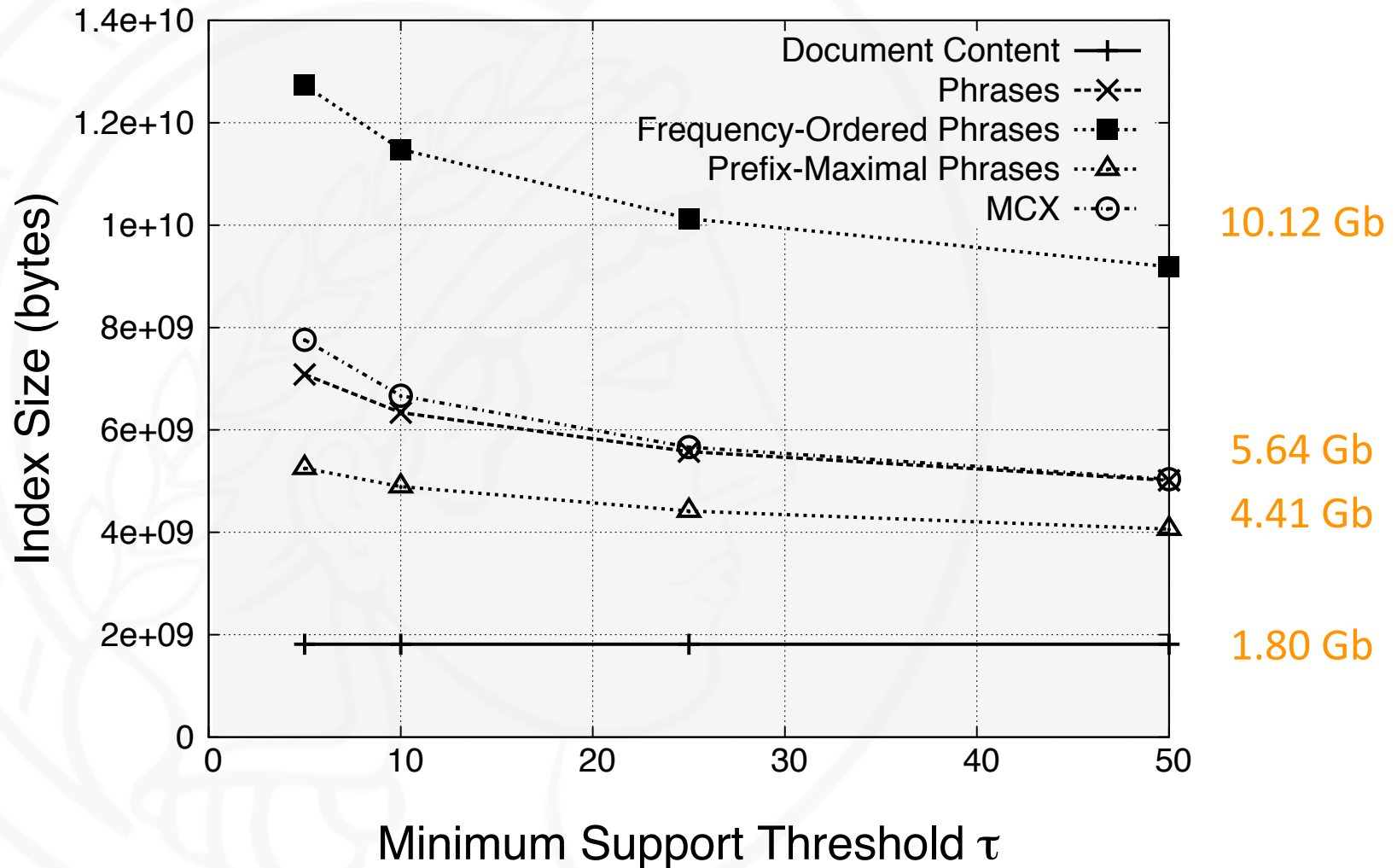
# Index Sizes for Different Values of $\tau$



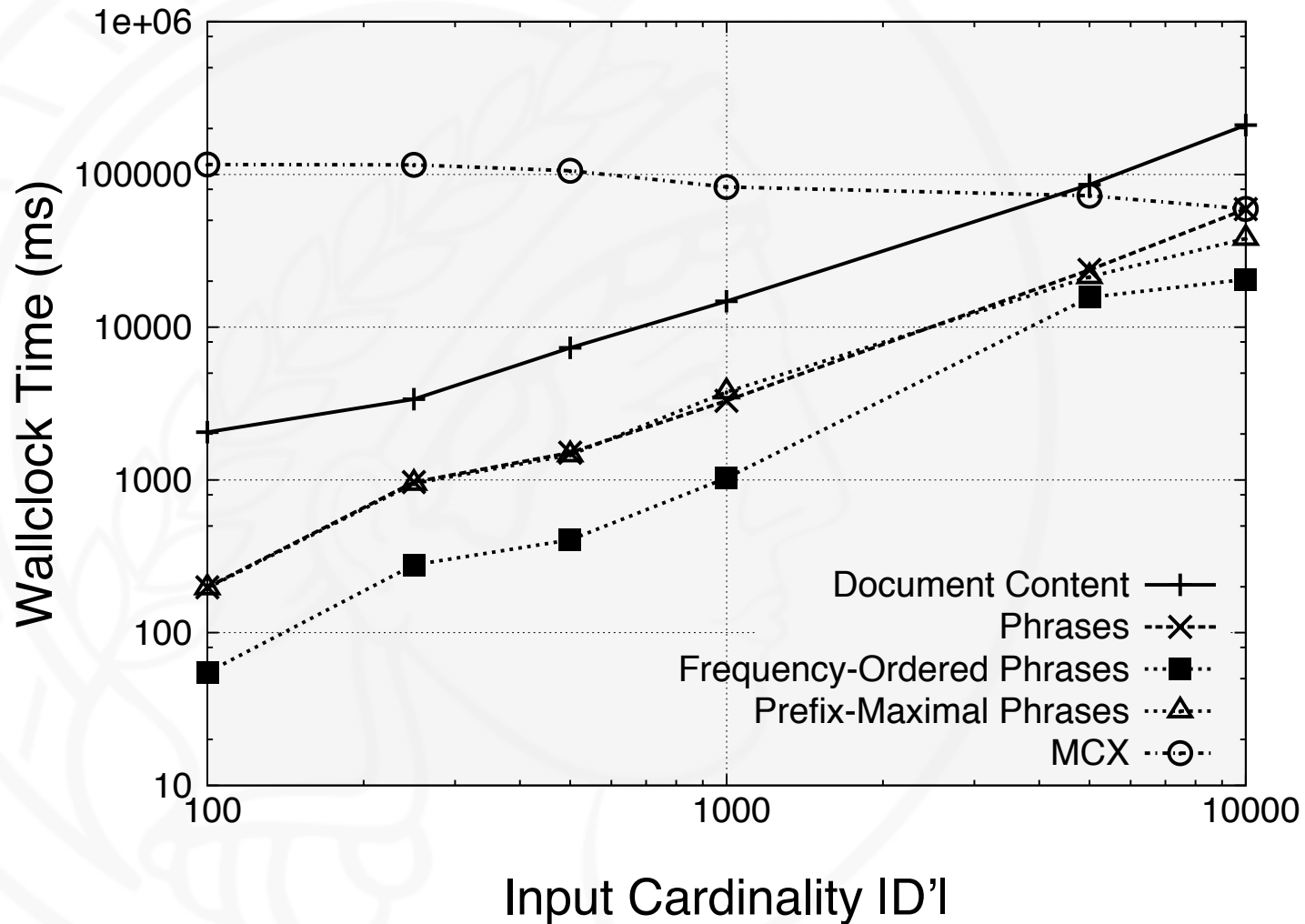
# Index Sizes for Different Values of $\tau$



# Index Sizes for Different Values of $\tau$



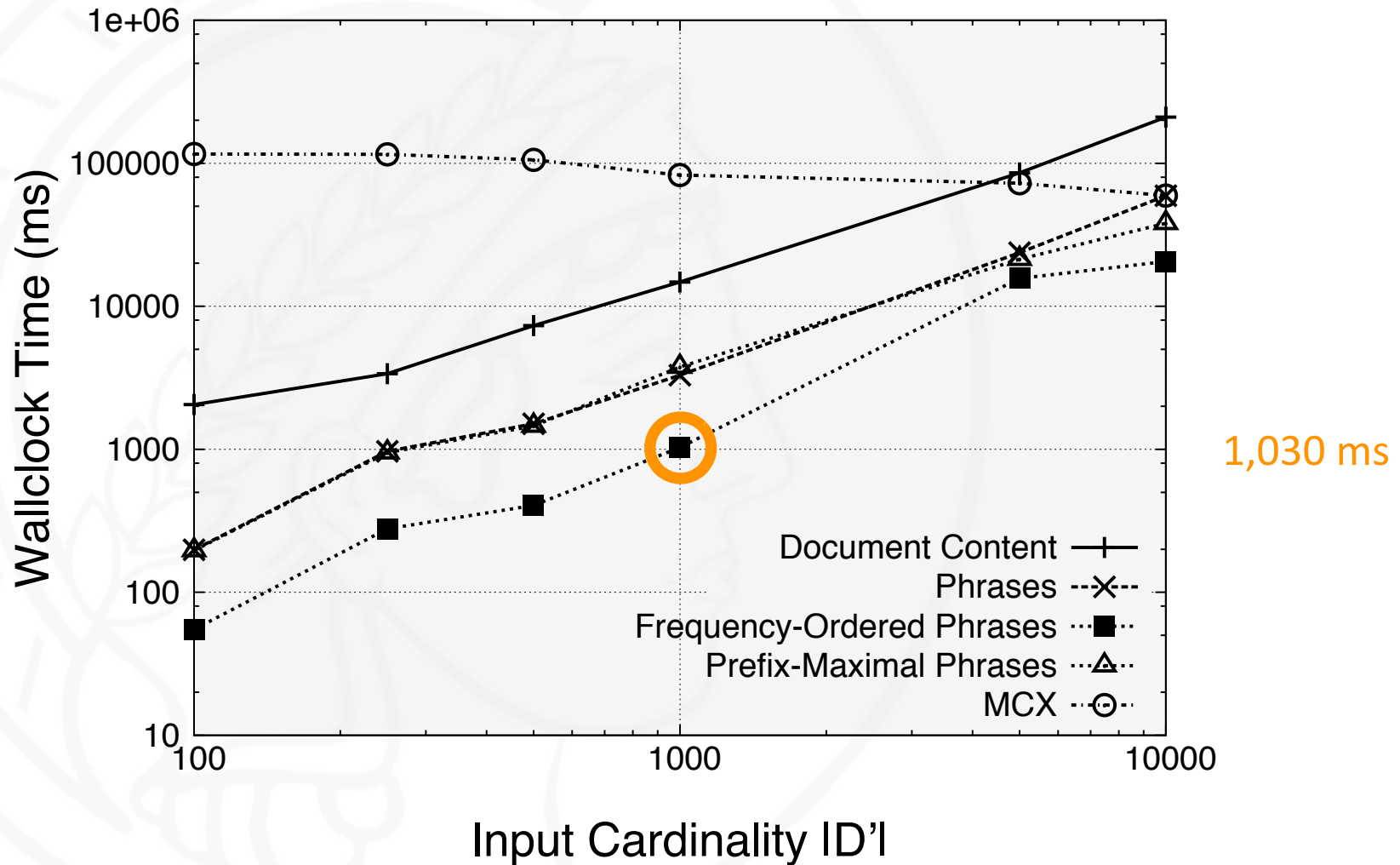
# Wall-Clock Times for Different Values of $|D'|$



$k = 100$   
 $\tau = 10$



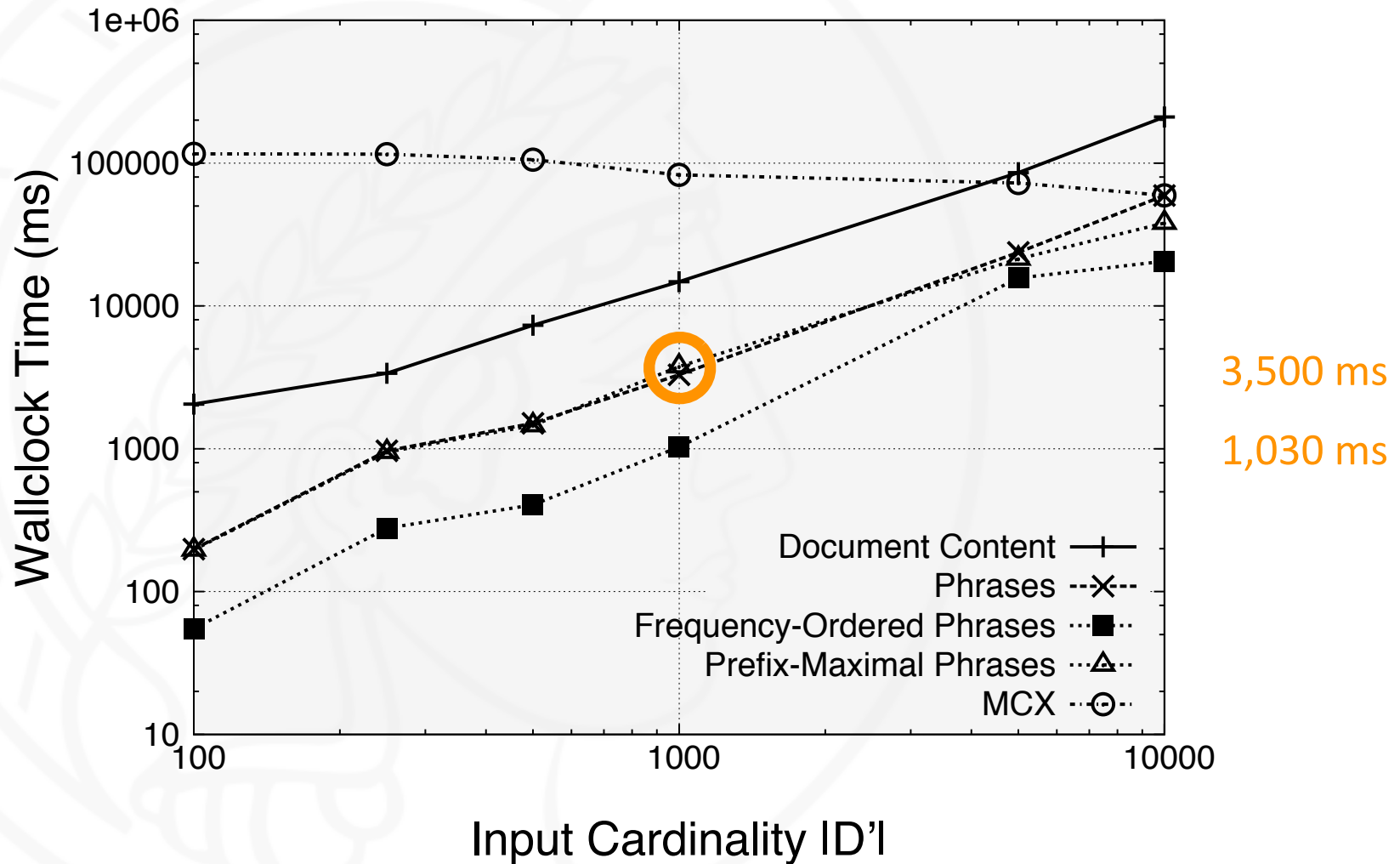
# Wall-Clock Times for Different Values of $|D'|$



$k = 100$   
 $\tau = 10$



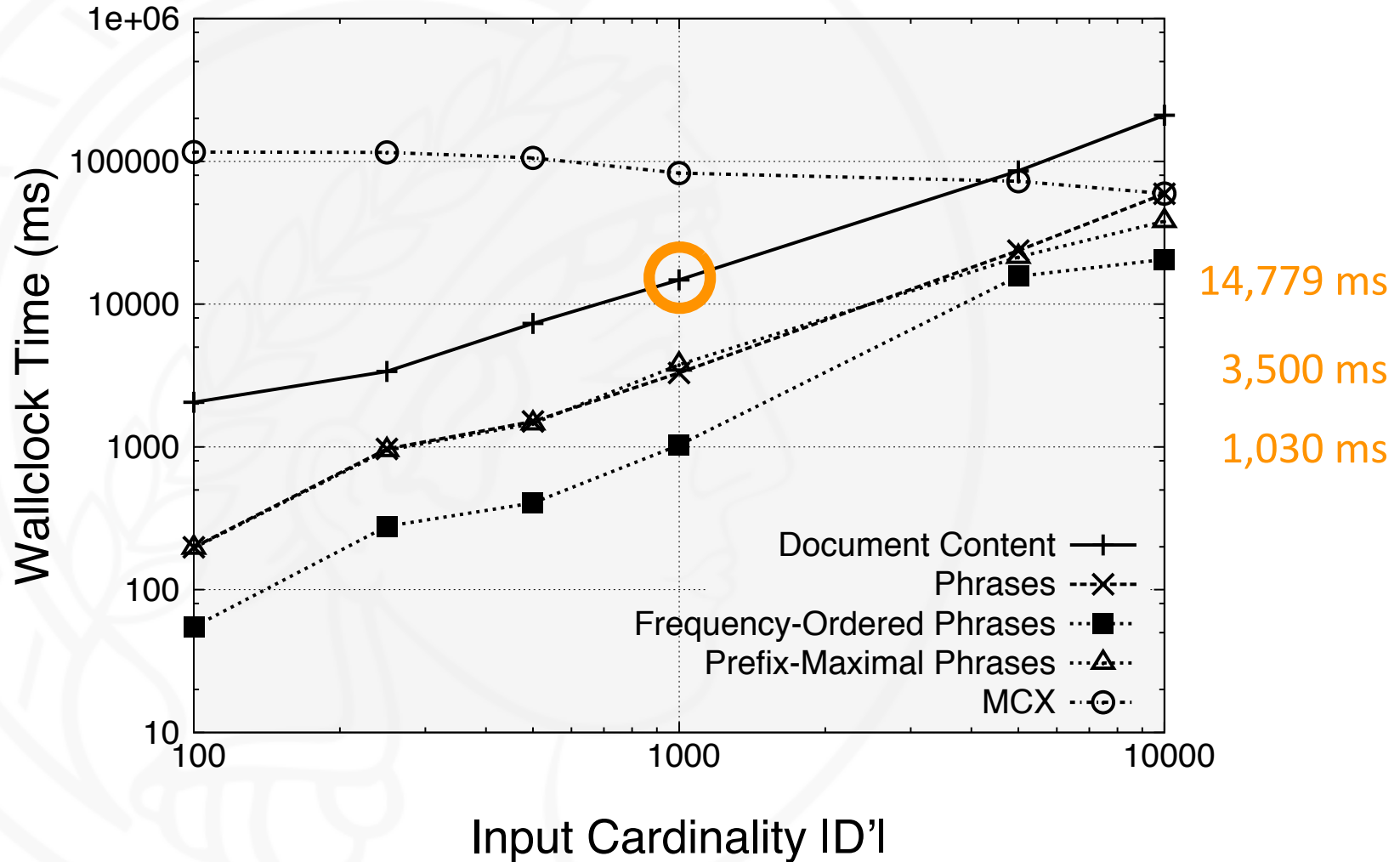
# Wall-Clock Times for Different Values of $|D'|$



$k = 100$   
 $\tau = 10$



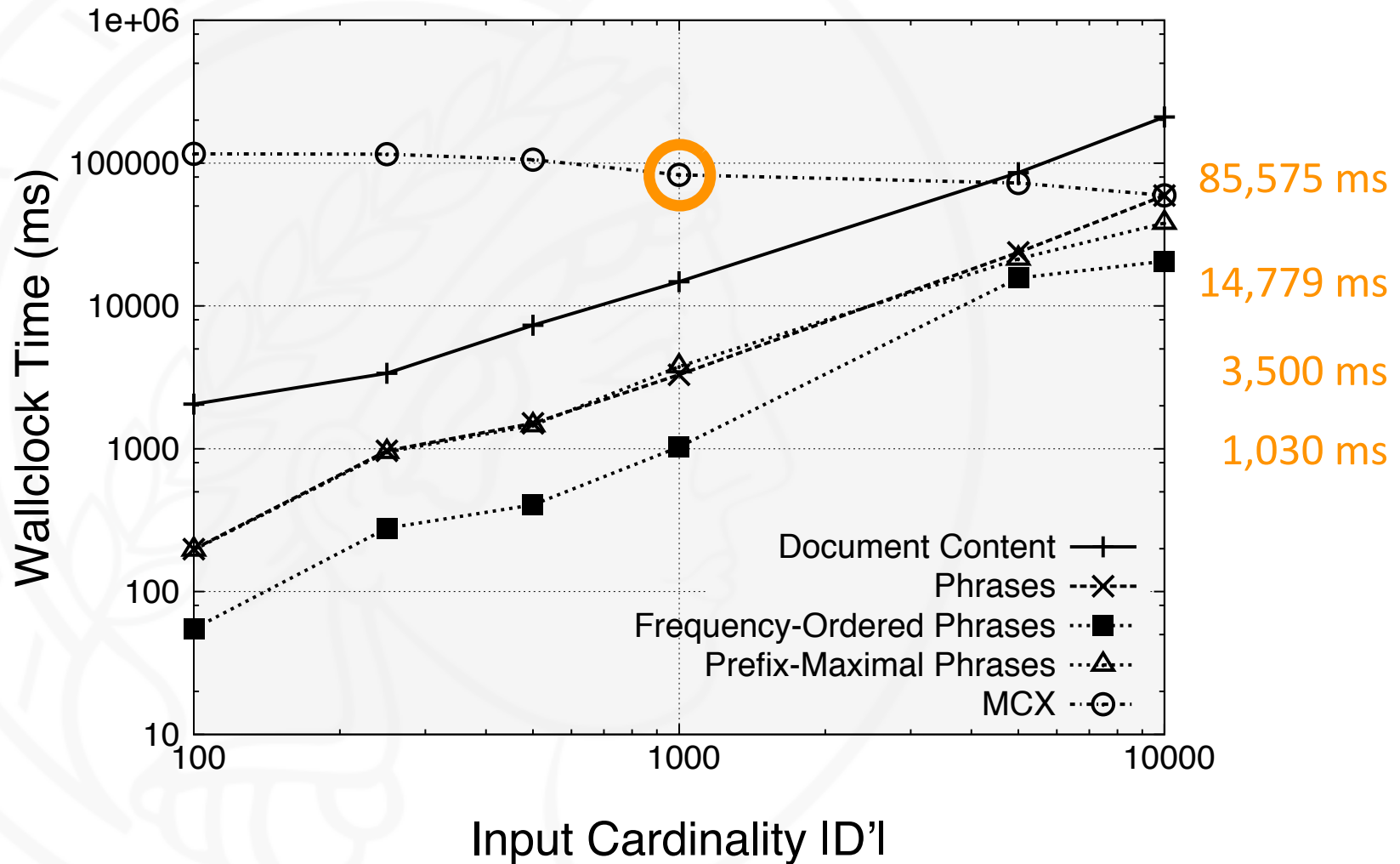
# Wall-Clock Times for Different Values of $|D'|$



$k = 100$   
 $\tau = 10$



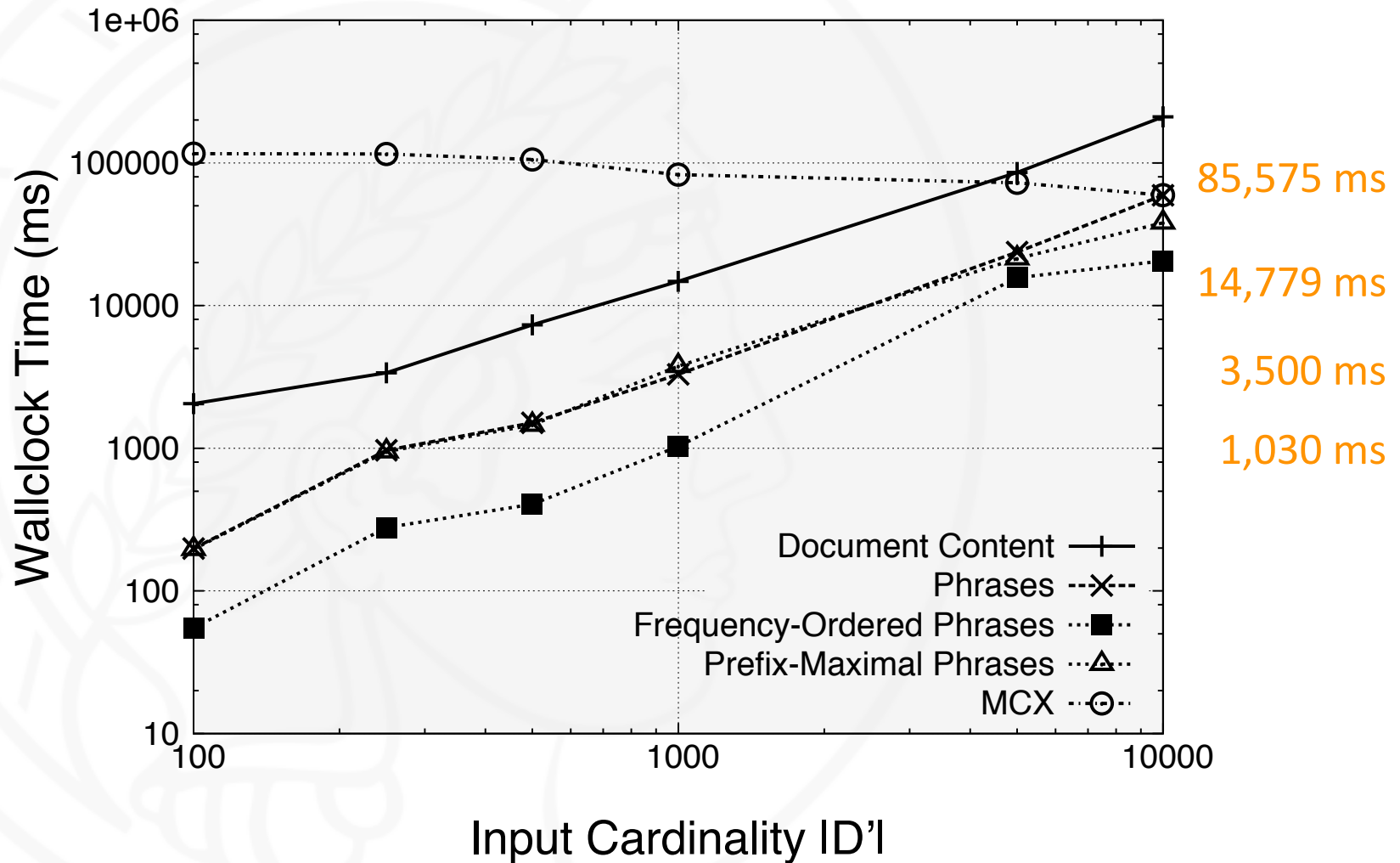
# Wall-Clock Times for Different Values of $|D'|$



$k = 100$   
 $\tau = 10$



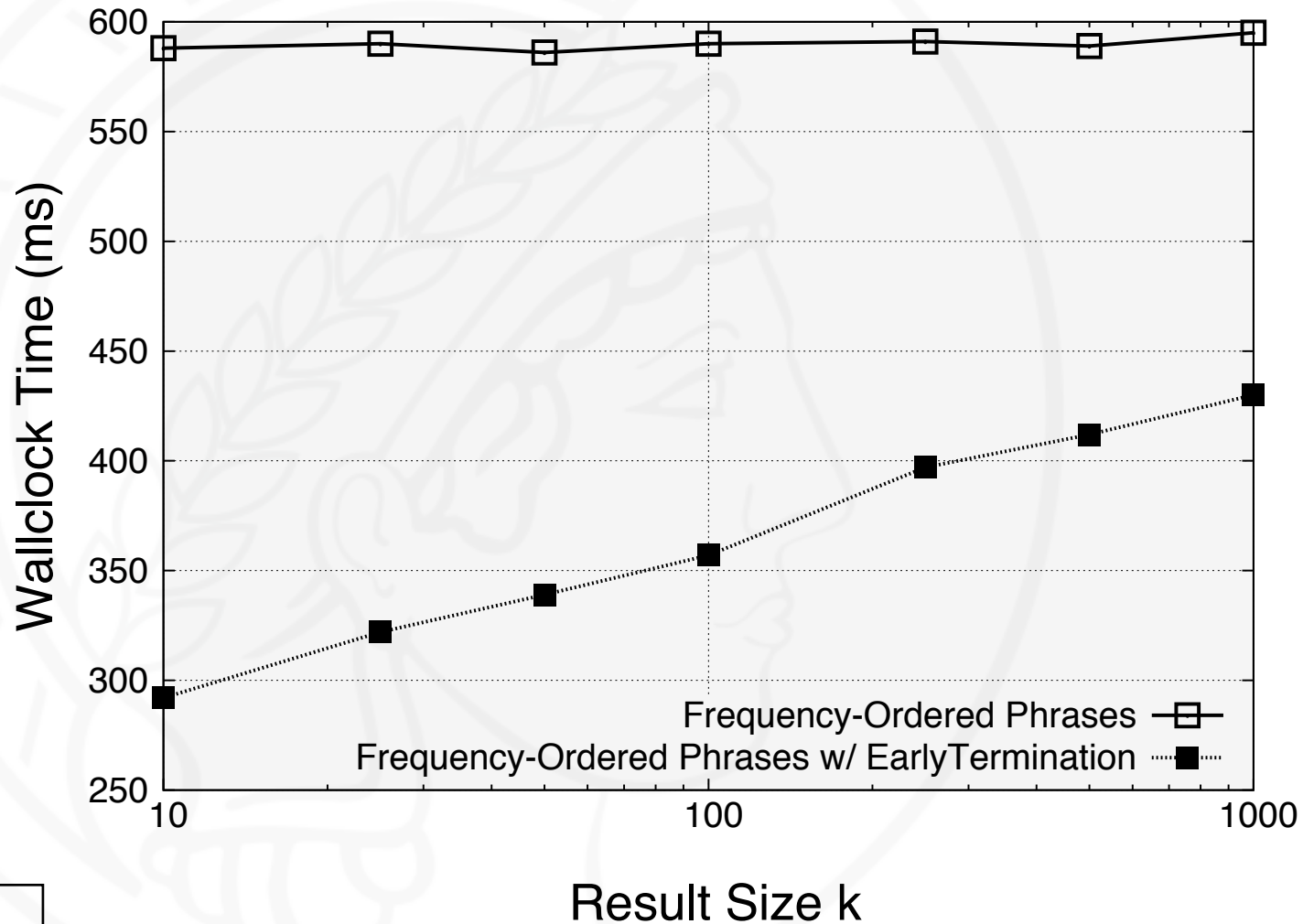
# Wall-Clock Times for Different Values of $|D'|$



$k = 100$   
 $\tau = 10$



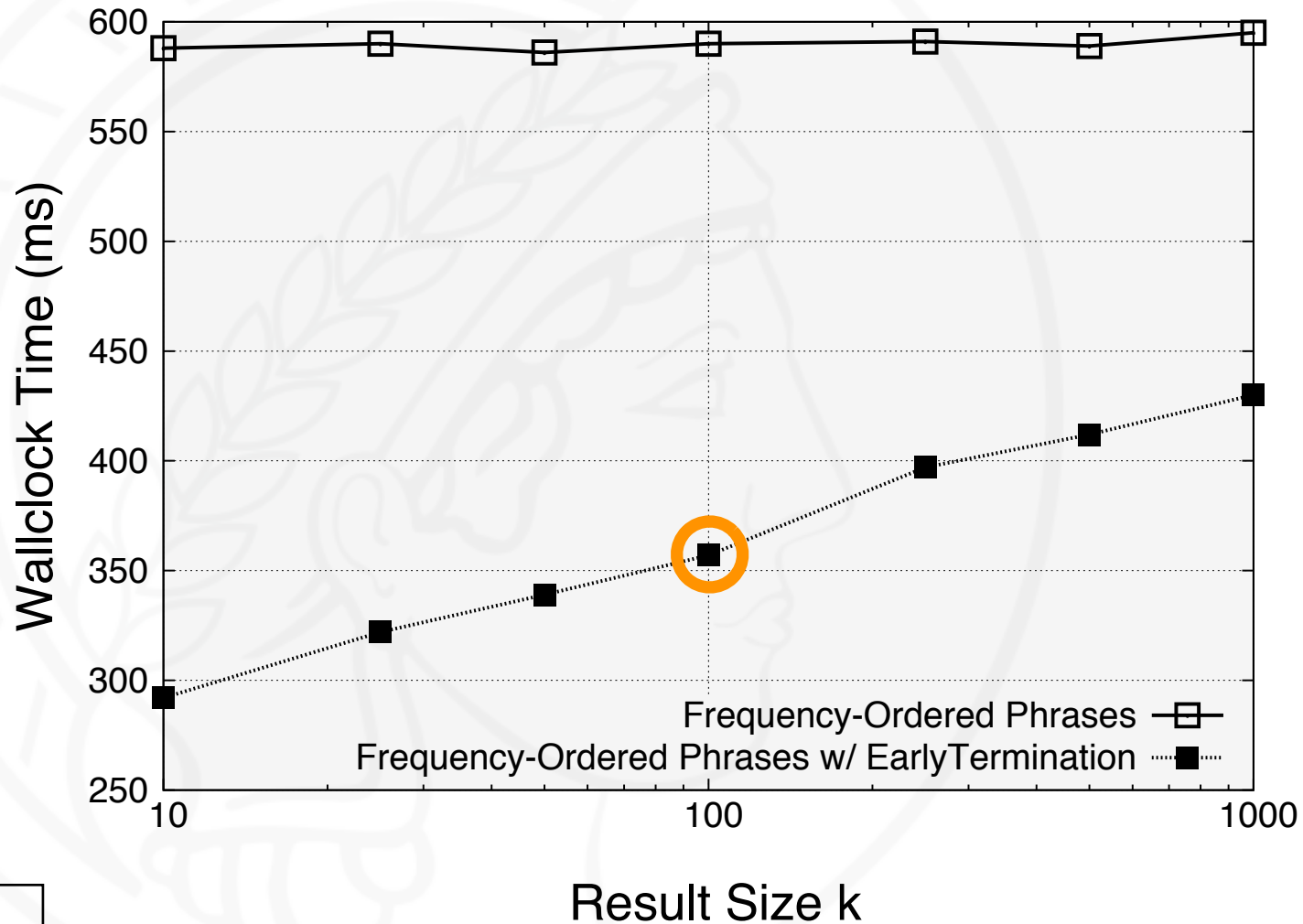
# Wall-Clock Times for Different Values of $k$



$\tau = 10$   
 $|D'| = 500$



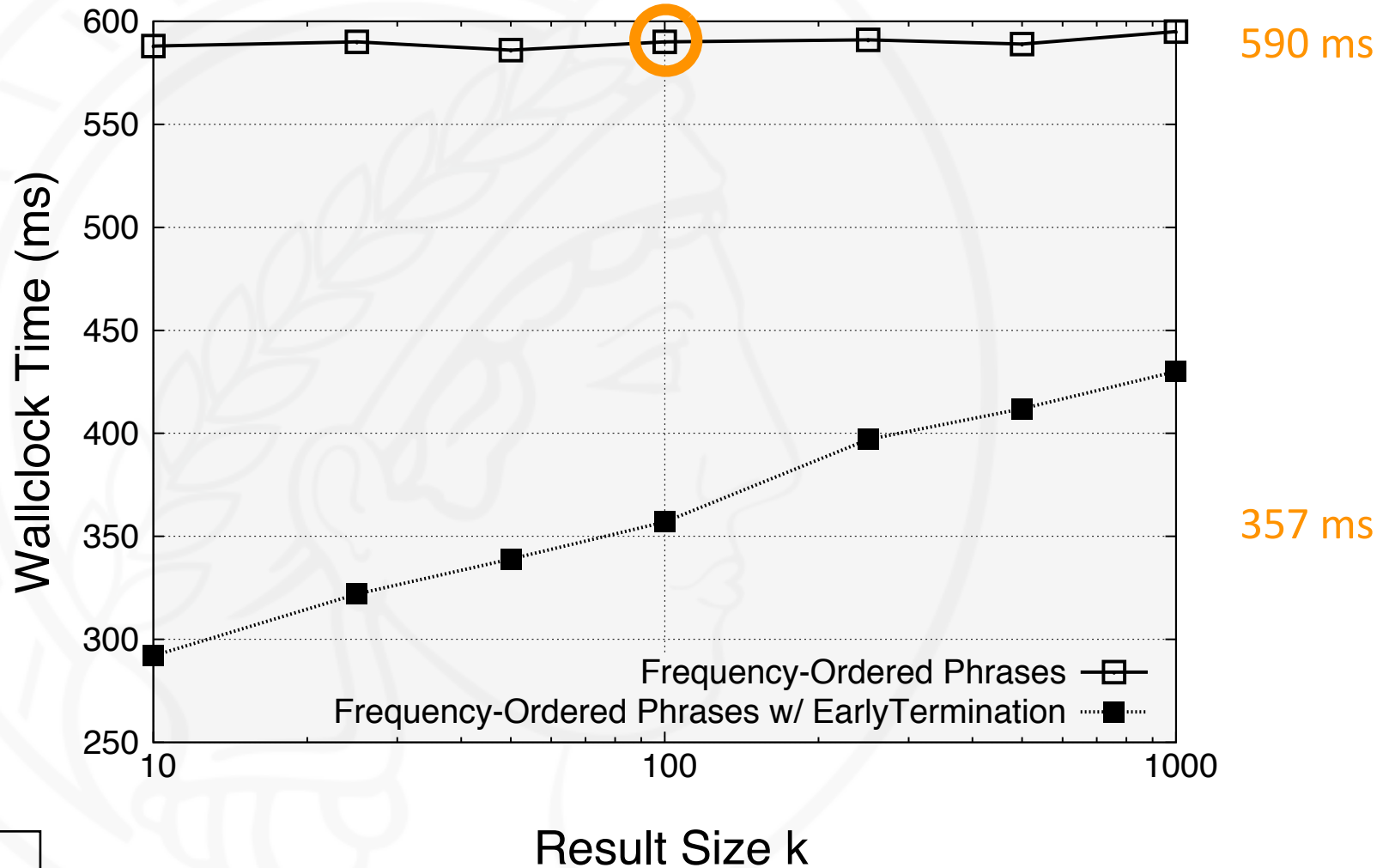
# Wall-Clock Times for Different Values of $k$



$\tau = 10$   
 $|D'| = 500$



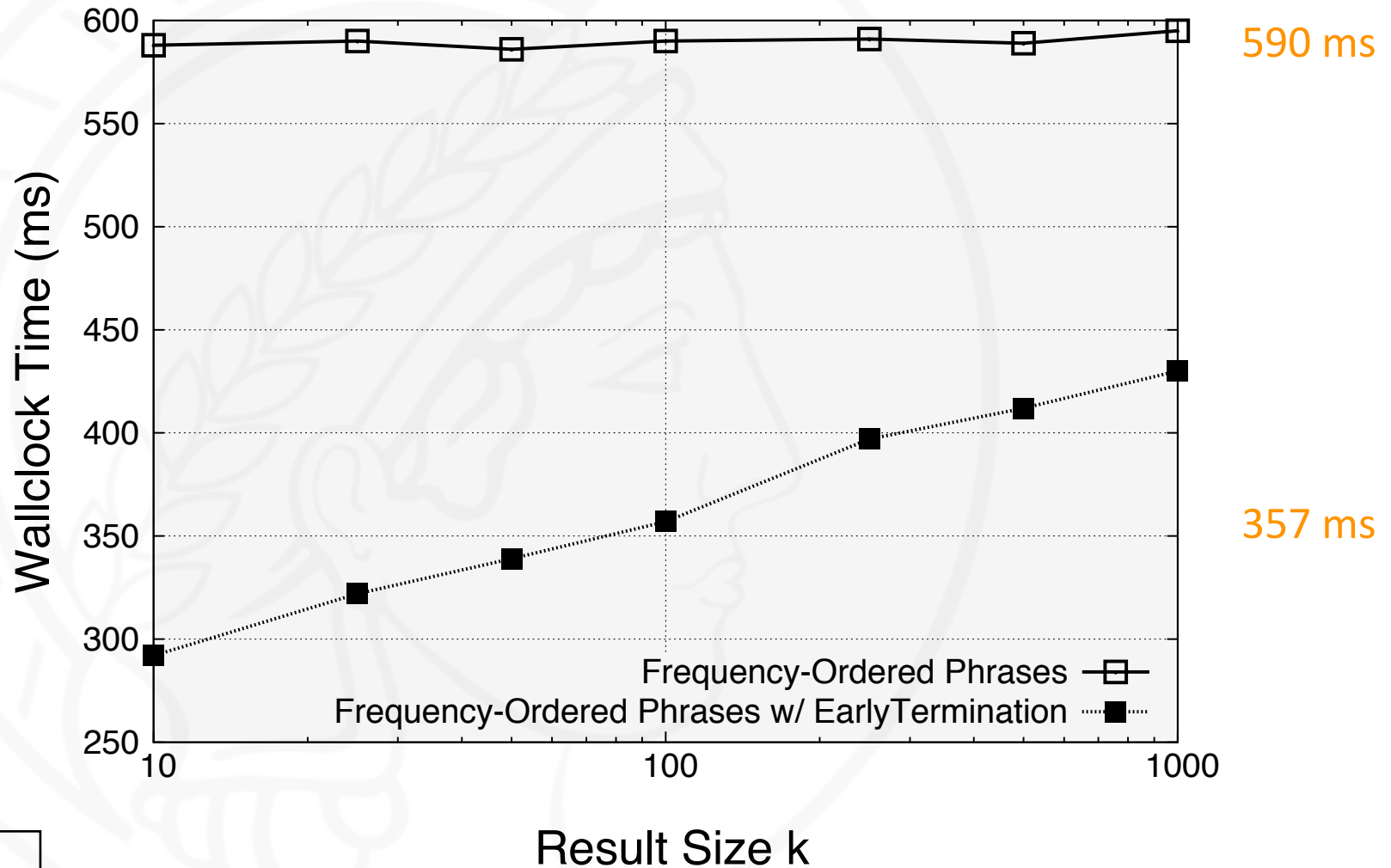
# Wall-Clock Times for Different Values of $k$



$\tau = 10$   
 $|D'| = 500$



# Wall-Clock Times for Different Values of $k$



$\tau = 10$   
 $|D'| = 500$



# Outline

- Motivation
- Problem Statement
- Our Approach
- Prior Art
- Experimental Evaluation
- Conclusion & Ongoing Research



# Conclusion


- Efficient identification of **interesting phrases** on **ad-hoc document sets** is a challenging research problem
- Our methods to tackle this problem
  - are based on **forward indexes**
  - differ in how they **represent document contents**
  - **Frequency-Ordered Phrases** allow for early termination
  - **Prefix-Maximal Phrases** result in a very compact index
- Experiments on real-world dataset show that our methods **work in practice** and **outperform the state-of-the-art method substantially** on realistic inputs



# Ongoing Research

- **Alternative scenario:** Identify phrases that best distinguish **two ad-hoc document sets** (e.g., **barack obama** vs. **george bush** or recently published vs. all documents on **UEFA**)
- Implement our index structures and pre-computations using **Hadoop** (or another MapReduce implementation)
- Support grouping of **almost-identical phrases** (e.g., **george bush said** vs. **george w. bush said**)
- Make use of **NLP techniques** such as part-of-speech tagging (e.g., to abstract from concrete pronouns)

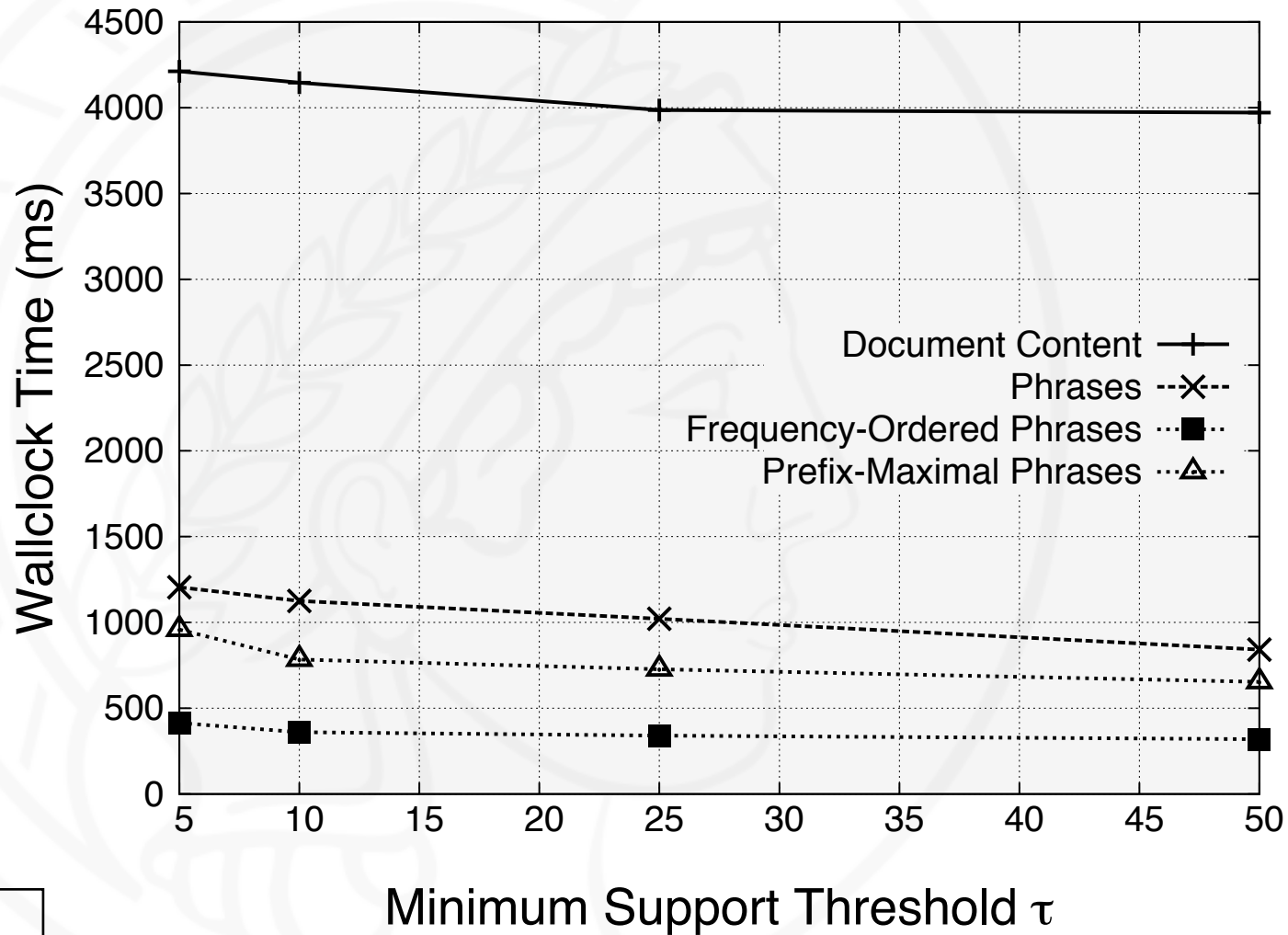




Thank you!  
Questions?



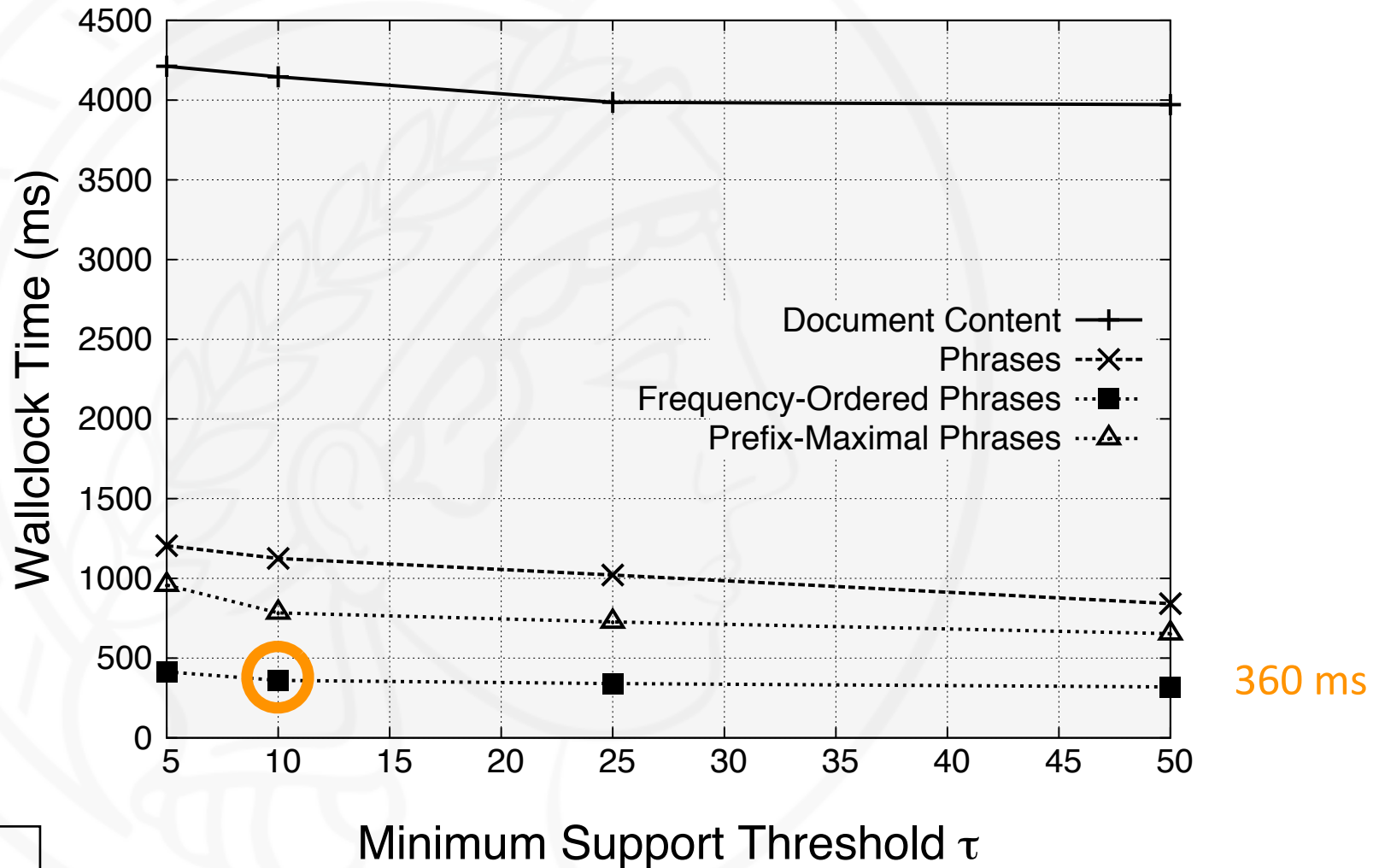
# Wall-Clock Times for Different Values of $\tau$



$k = 100$   
 $|D'| = 500$



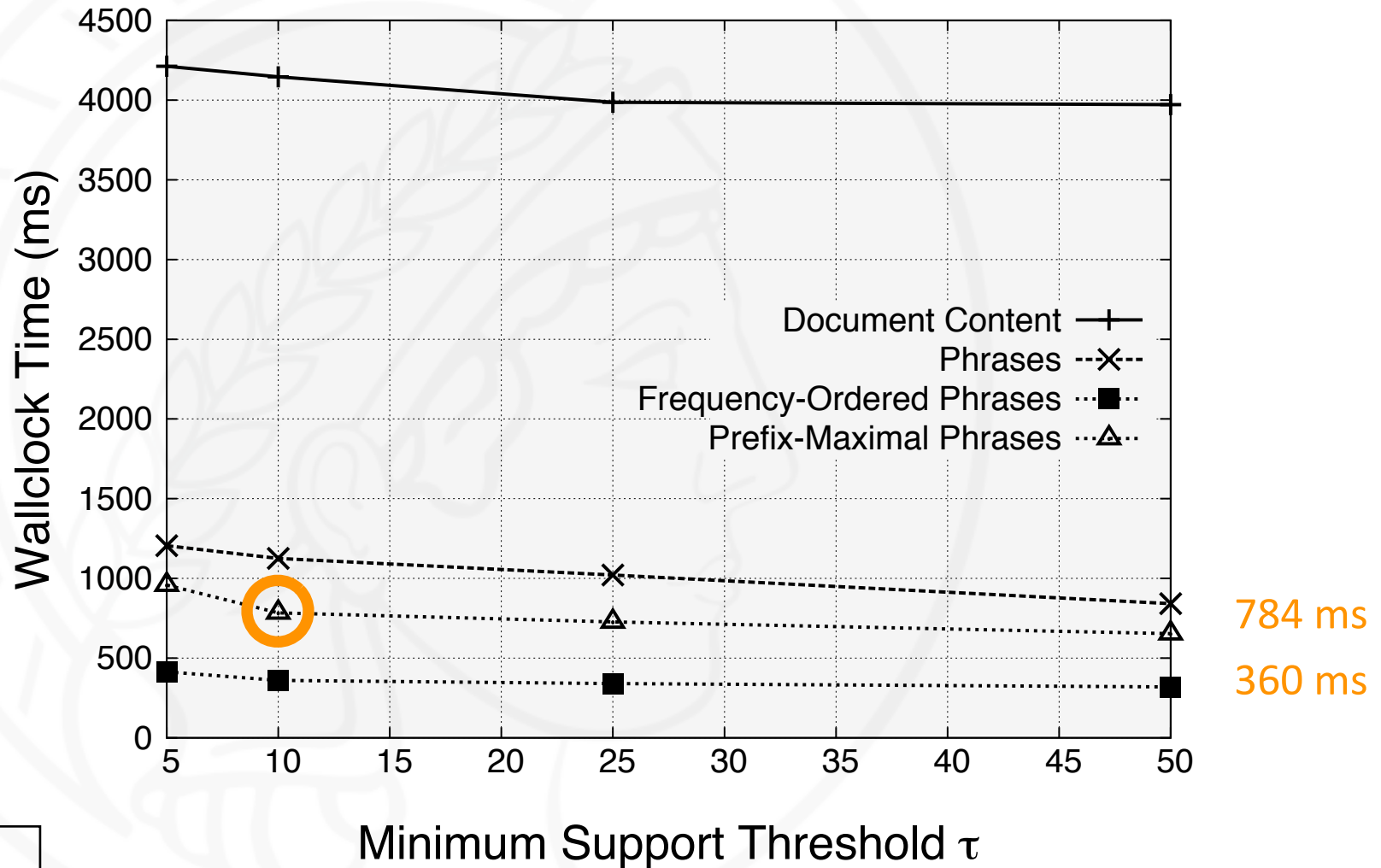
# Wall-Clock Times for Different Values of $\tau$



$k = 100$   
 $|D'| = 500$



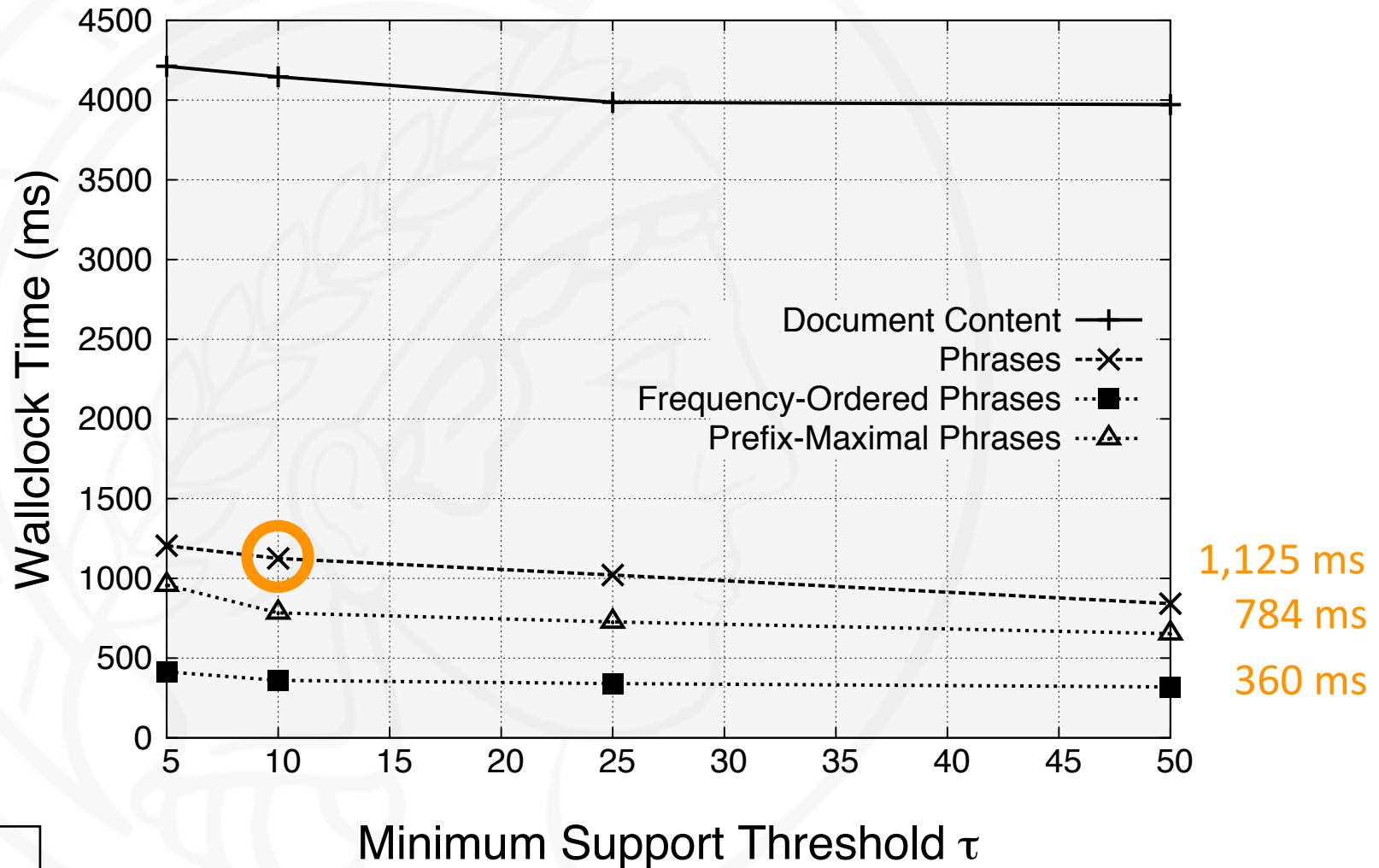
# Wall-Clock Times for Different Values of $\tau$



$k = 100$   
 $|D'| = 500$



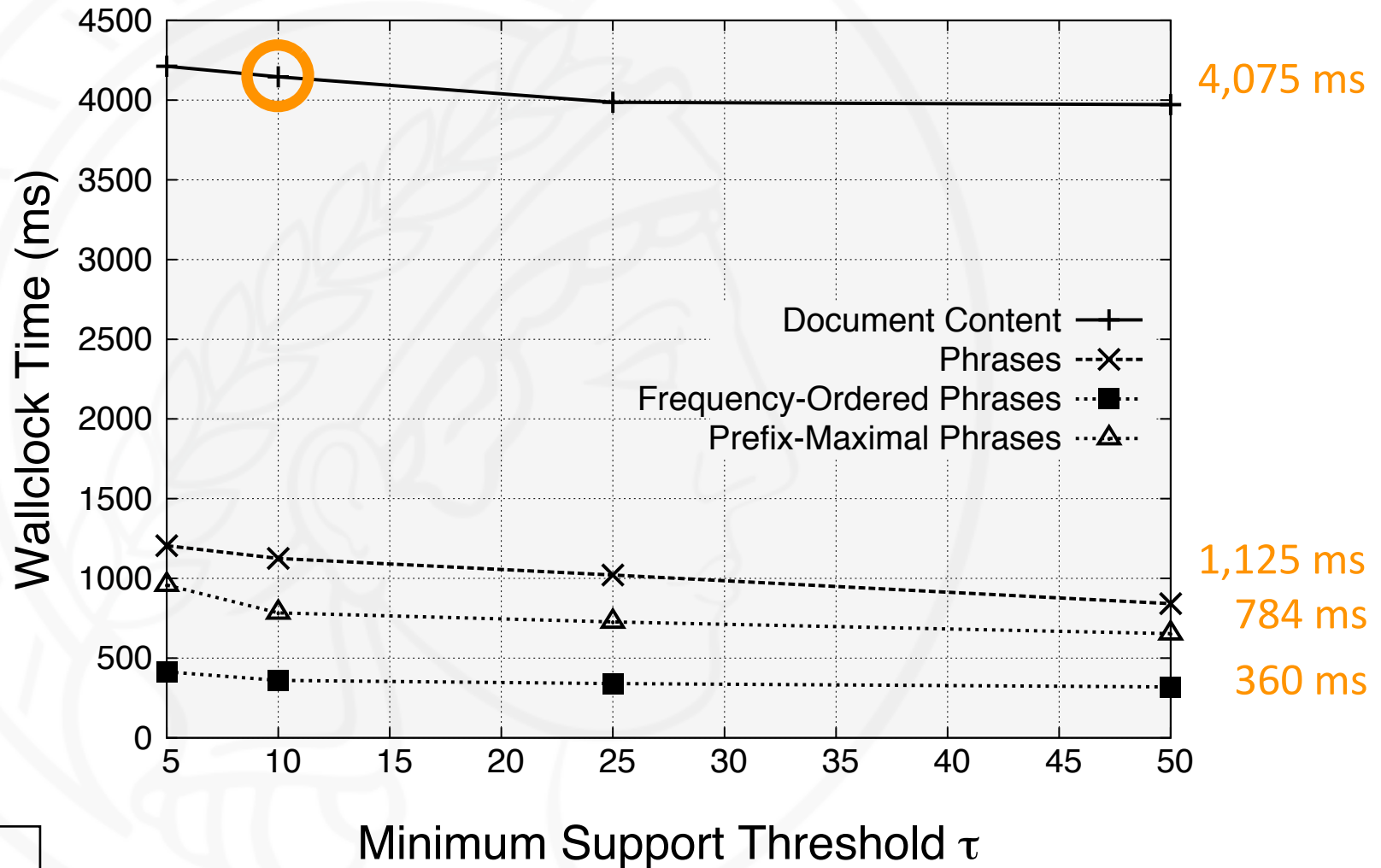
# Wall-Clock Times for Different Values of $\tau$



$k = 100$   
 $|D'| = 500$



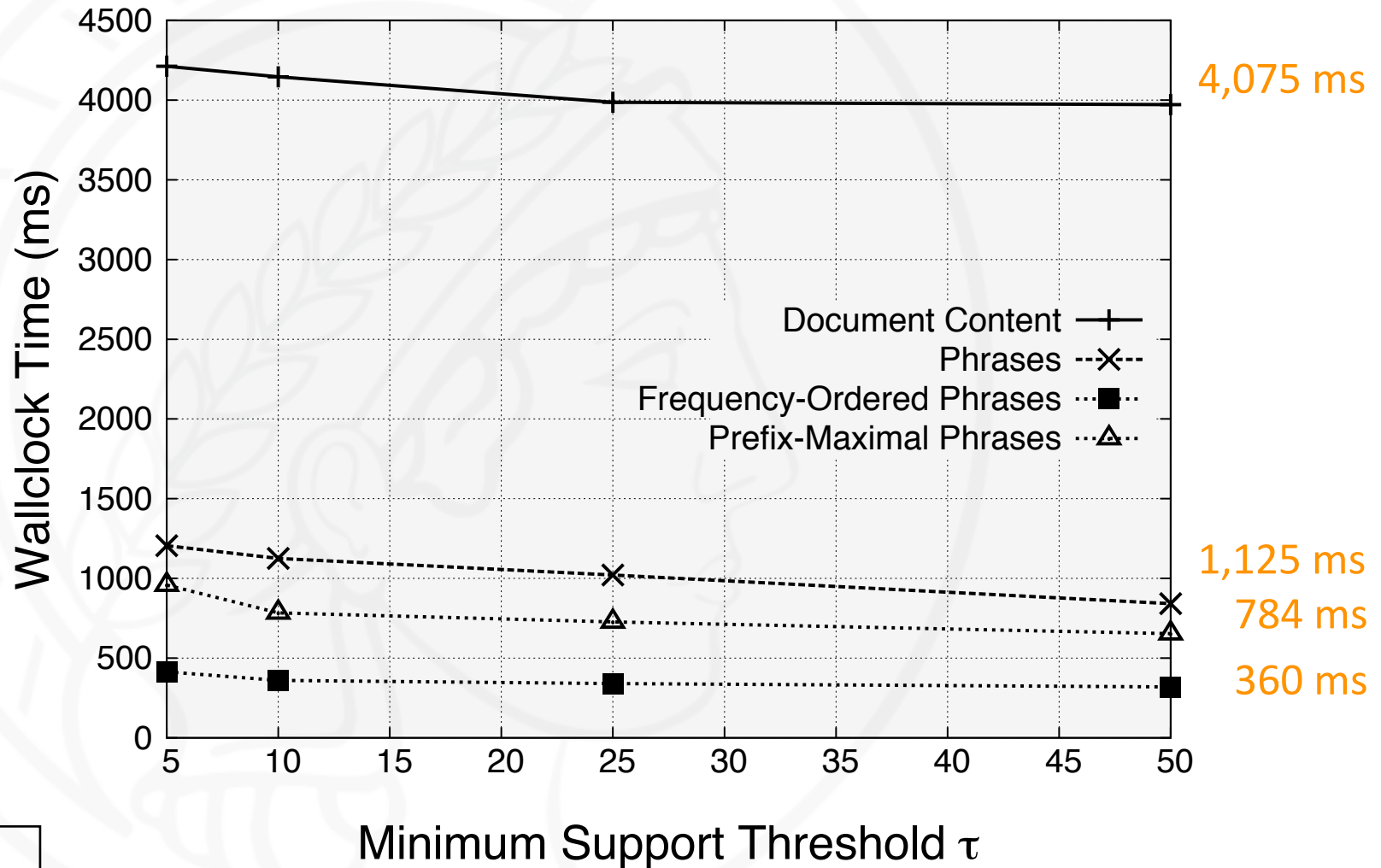
# Wall-Clock Times for Different Values of $\tau$



$k = 100$   
 $|D'| = 500$



# Wall-Clock Times for Different Values of $\tau$



$k = 100$   
 $|D'| = 500$

