

Indexing Boolean Expressions

Steven Whang**, Chad Brower*, Jayavel
Shanmugasundaram*, Sergei Vassilvitskii*, Erik Vee*,
Ramana Yerneni*, Hector Garcia-Molina**

*Yahoo! Research

**Stanford University

YAHOO!



Problem

- Ad Example
 - BE: $\text{age} \in \{10,20\}$ & $\text{country} \notin \{\text{US}\}$
 - S: $\text{age}=20$ & $\text{country}=\text{FR}$ & $\text{gender}=\text{F}$



Problem

- Ad Example
 - BE: $\text{age} \in \{10,20\} \ \& \ \text{country} \notin \{\text{US}\}$
 - S: $\text{age}=20 \ \& \ \text{country}=\text{FR} \ \& \ \text{gender}=\text{F}$
- Given an assignment S, find all matching boolean expressions (BEs)

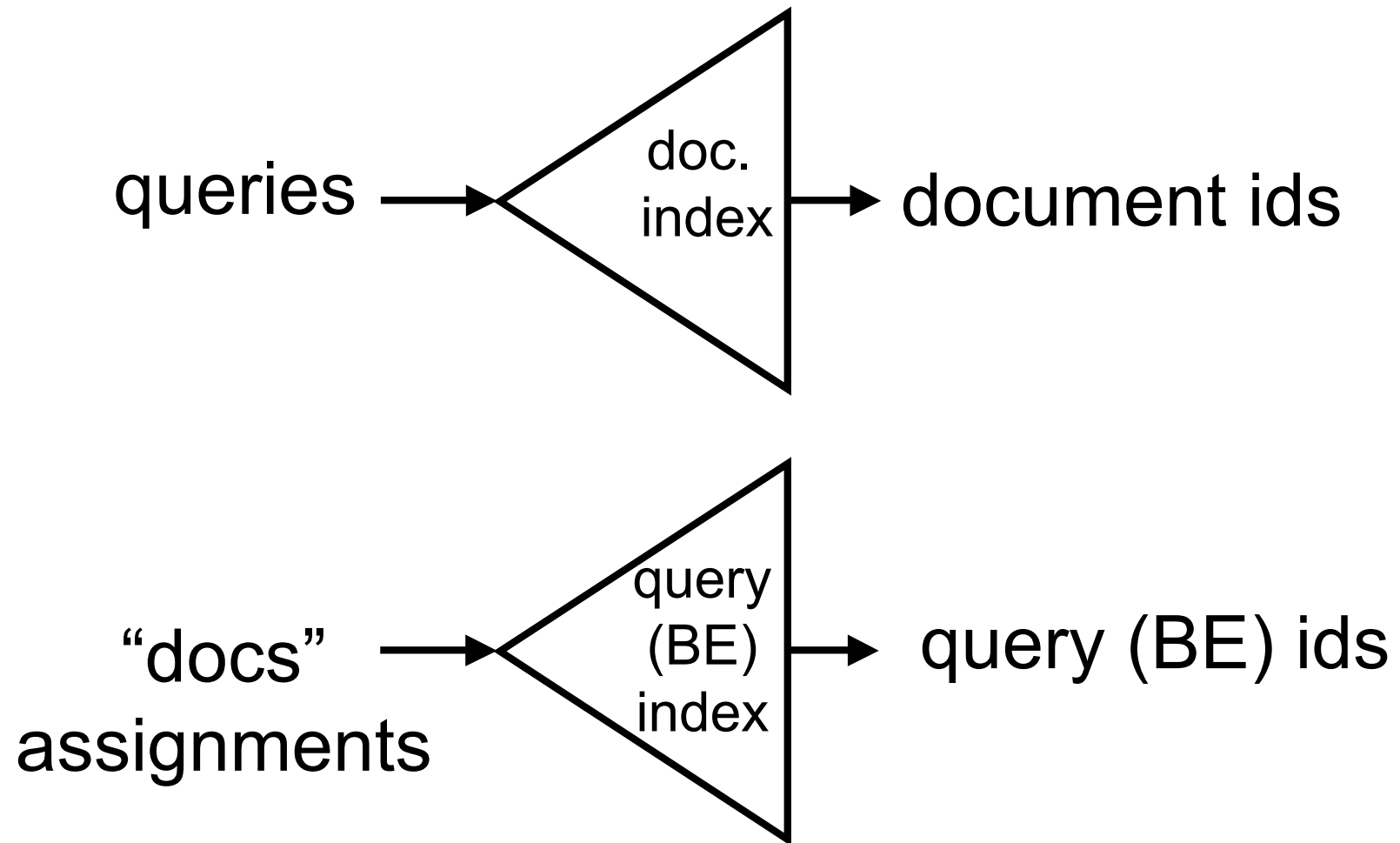


Possible applications

- Display advertising
- Publish/subscribe systems

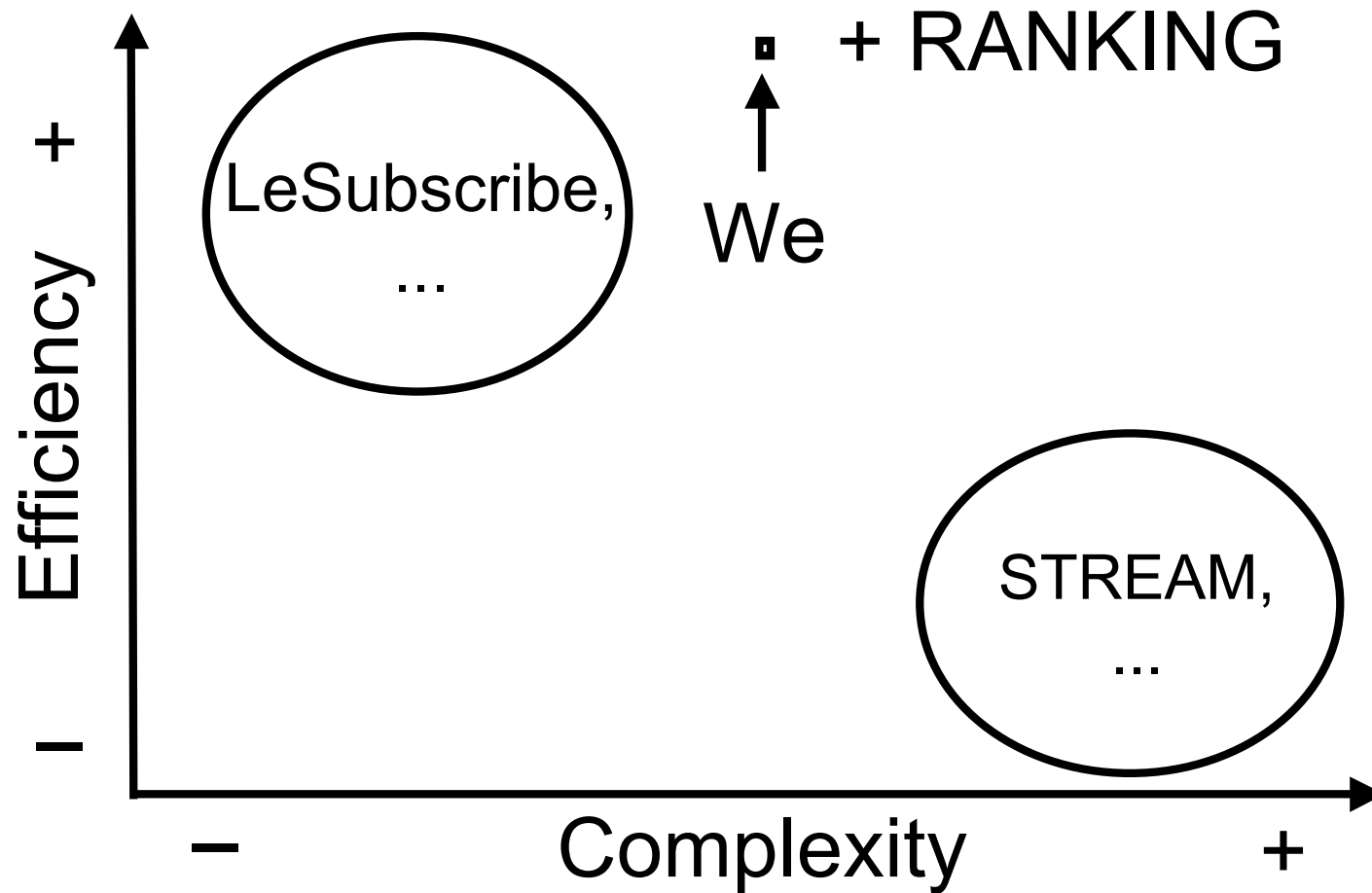


Inverted index






Related works





Contributions

- Use inverted indexing techniques for “complex” BEs
 - DNF, CNF expressions of \in , \notin predicates with multiple values
- Support top-k pruning given relevance scores

- Inverted index construction 
- Search algorithms
 - Conjunction (\in only)
 - DNF
 - CNF (\in only)
- Experimental results



Inverted index

E1: $A \in \{1\}$

E2: $A \in \{1\} \ \& \ B \in \{2\} \ \& \ C \in \{3,4\}$

Key	Posting List
(A,1)	E1, E2
(B,2)	E2
(C,3)	E2
(C,4)	E2



Inverted index

E1: $A \in \{1\}$

E2: $A \in \{1\} \ \& \ B \in \{2\} \ \& \ C \in \{3,4\}$

S: $A=1 \ \& \ B=2$


Key	Posting List
(A,1)	E1, E2
(B,2)	E2
(C,3)	E2
(C,4)	E2

} Posting
Lists
Used



Inverted index

- Conventional posting list operations (intersection, union) do not work correctly

- Inverted index construction
- Search algorithms
 - Conjunction (\in only) 
 - DNF
 - CNF (\in only)
- Experimental results



Conjunction algorithm (\in only)

S: A=1 & B=2 & C=3 & D=4

K=3

Key	Posting List
(A,1)	<u>E1</u> , E2, E3
(B,2)	<u>E2</u> , E3
(C,3)	<u>E3</u> , E4
(D,4)	<u>E4</u>

Matching conjunctions of size K: $\{\}$



Conjunction algorithm (\in only)

S: A=1 & B=2 & C=3 & D=4

K=3

Key	Posting List
(A,1)	E1, E2, <u>E3</u>
(B,2)	E2, <u>E3</u>
(C,3)	<u>E3</u> , E4
(D,4)	<u>E4</u>

Matching conjunctions of size K: $\{\}$



Conjunction algorithm (\in only)

S: A=1 & B=2 & C=3 & D=4

K=3

Key	Posting List
(A,1)	E1, E2, <u>E3</u>
(B,2)	E2, <u>E3</u>
(C,3)	<u>E3</u> , E4
(D,4)	<u>E4</u>

Matching conjunctions of size K: {E3}



Conjunction algorithm (\in only)

S: A=1 & B=2 & C=3 & D=4

K=3

Key	Posting List
(A,1)	E1, E2, E3 (end of list)
(B,2)	E2, E3 (end of list)
(C,3)	E3, <u>E4</u>
(D,4)	<u>E4</u>

Matching conjunctions of size K: {E3}



Conjunction algorithm (\in only)

S: A=1 & B=2 & C=3 & D=4

K=3

Key	Posting List
(C,3)	E3, <u>E4</u>
(D,4)	<u>E4</u>
(A,1)	E1, E2, E3 (end of list)
(B,2)	E2, E3 (end of list)

Matching conjunctions of size K: {E3}



Conjunction algorithm (\in only)


S: A=1 & B=2 & C=3 & D=4

K=3

Key	Posting List
(C,3)	E3, E4 (end of list)
(D,4)	E4 (end of list)
(A,1)	E1, E2, E3 (end of list)
(B,2)	E2, E3 (end of list)

Matching conjunctions of size K: {E3}

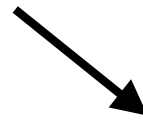
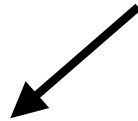
Outline

- Inverted index construction
- Search algorithms
 - Conjunction (\in only)
 - DNF 
 - CNF (\in only)
- Experimental results



DNF algorithm

E1: $(A \in \{1,2\} \ \& \ B \in \{4\}) \mid$
 $(A \in \{3\} \ \& \ B \in \{5\})$



E1.1: $A \in \{1,2\} \ \&$
 $B \in \{4\}$

E1.2: $A \in \{3\} \ \&$
 $B \in \{5\}$



DNF algorithm

E1: $(A \in \{1,2\} \ \& \ B \in \{4\}) \mid$
 $(A \in \{3\} \ \& \ B \in \{5\})$

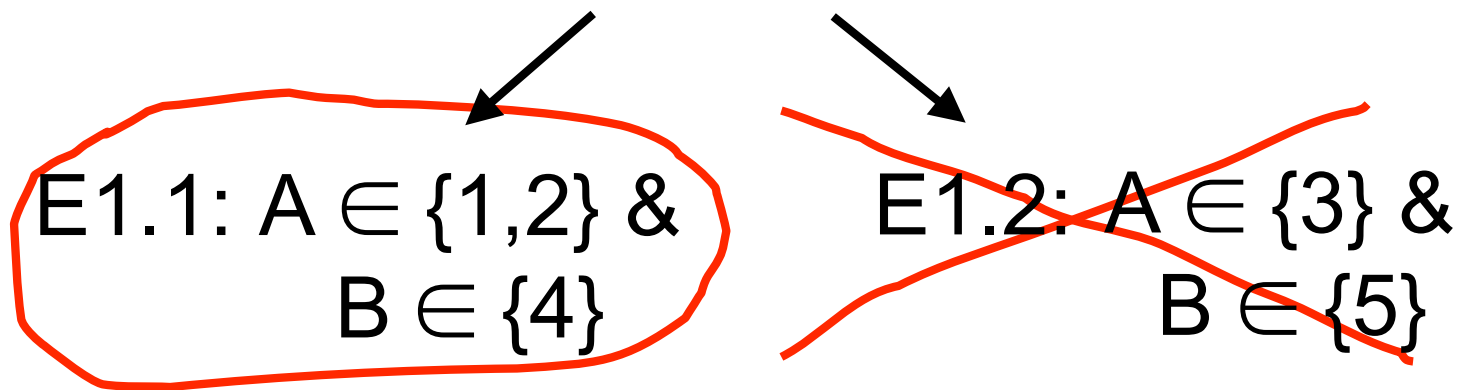
E1.1: $A \in \{1,2\} \ \&$
 $B \in \{4\}$

~~E1.2: $A \in \{3\} \ \&$
 $B \in \{5\}$~~




DNF algorithm

E1: $(A \in \{1,2\} \ \& \ B \in \{4\}) \mid$
 $(A \in \{3\} \ \& \ B \in \{5\})$



E1 is satisfied

Outline

- Inverted index construction
- Search algorithms
 - Conjunction (\in only)
 - DNF
 - CNF (\in only) 
- Experimental results



CNF algorithm (\in only)

- Similar to conjunction algorithm except:
 - Size of BE = # of disjunctions
 - Make sure all disjunctions of BE are satisfied using disjunction IDs

E1: $(A \in \{1\} \mid B \in \{2\}) \ \& \ (C \in \{3\} \mid D \in \{4\})$


disjunction 1 disjunction 2



More algorithms in paper

- Search algorithms for DNF/CNF BEs with \notin predicates
- Top-k algorithms for DNF/CNF BEs based on relevance scores

Outline

- Inverted index construction
- Search algorithms
 - Conjunction
 - DNF
 - CNF
- Experimental results 

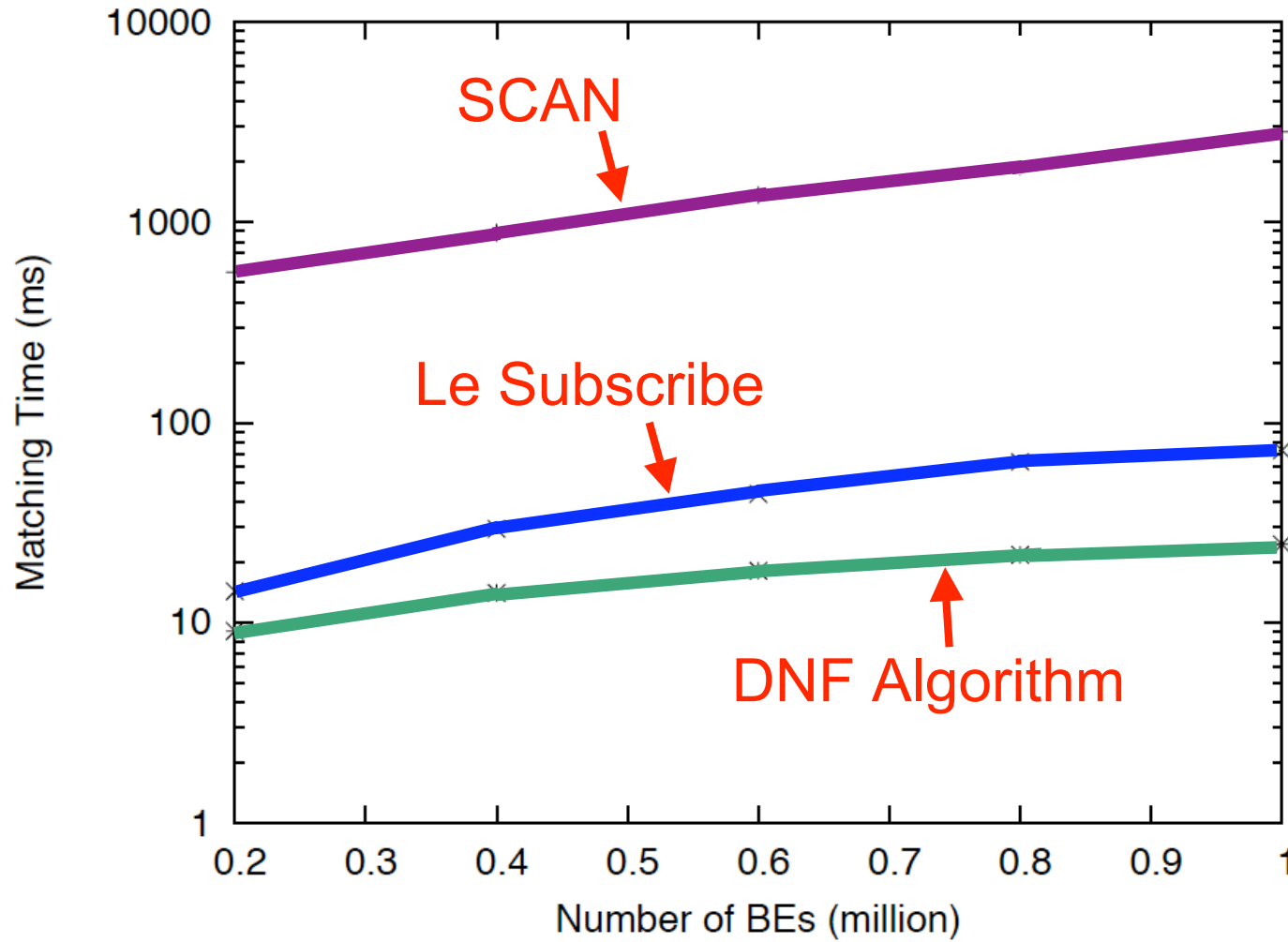


Data sets

- Assignments = Display advertising Impressions (Ad opportunities)
- BEs = Synthetic workloads generated from display advertising contracts
- Up to 1 million DNF/CNF BEs generated
- High dimensional (~1500)

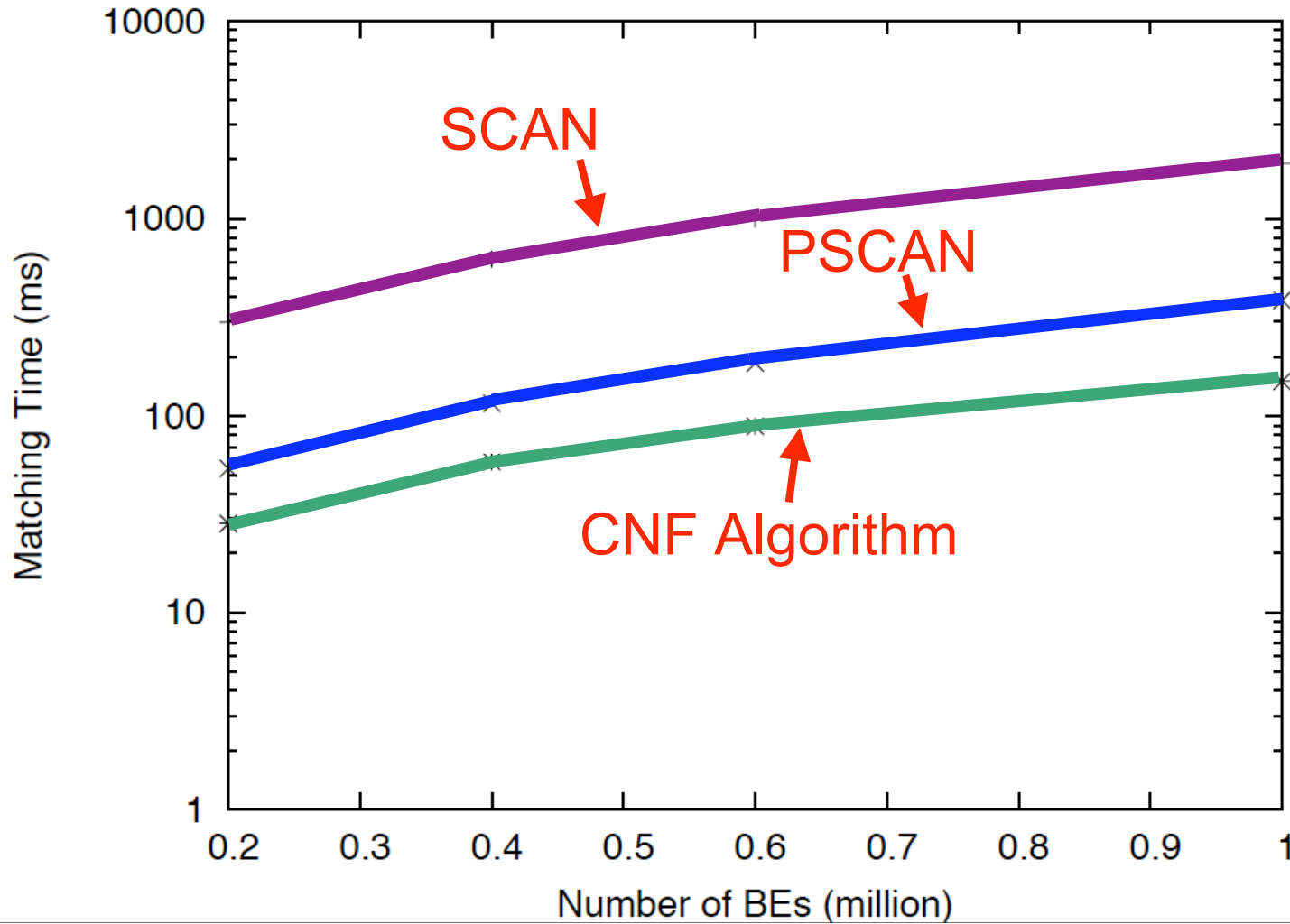


DNF algorithm scalability



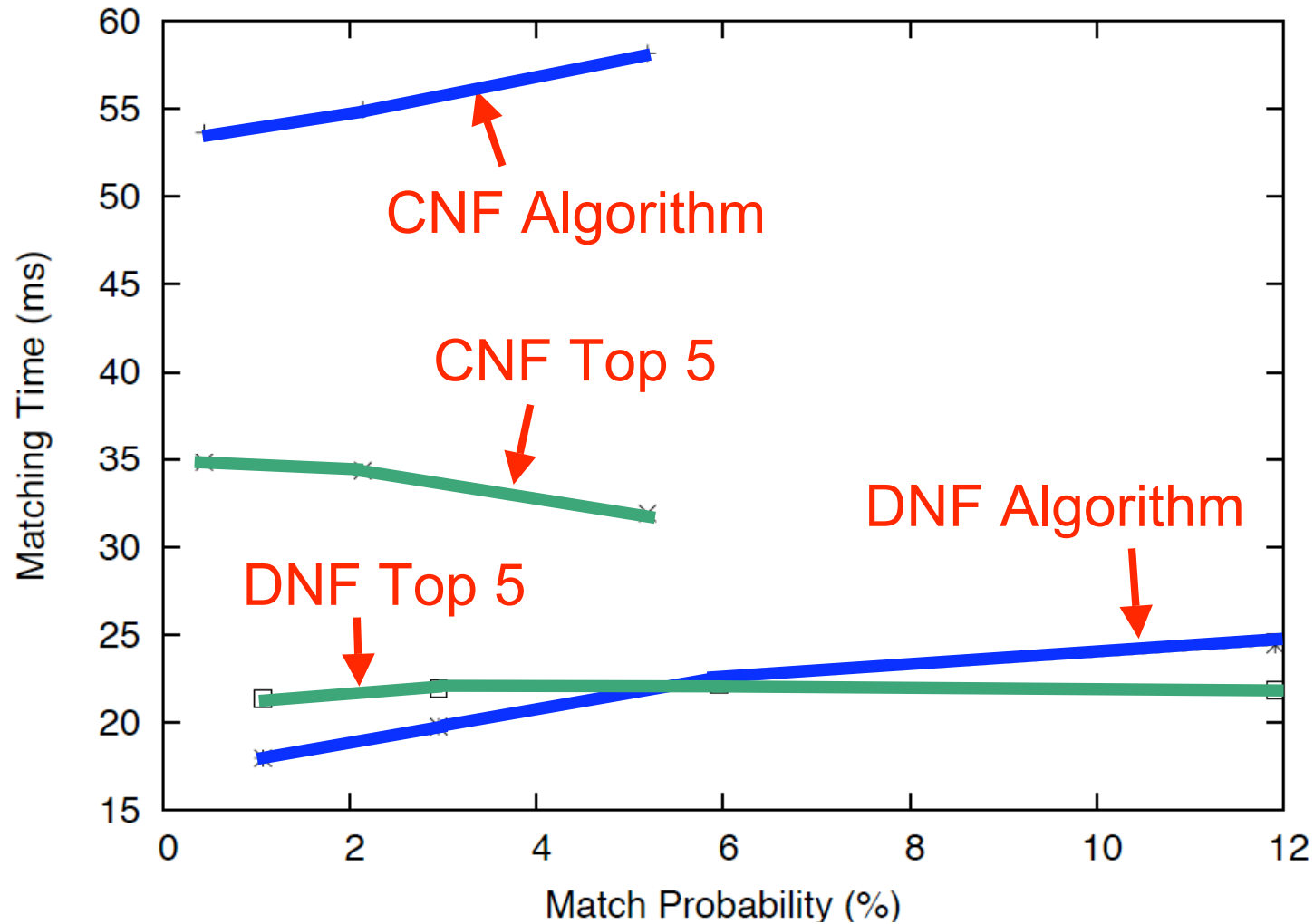


CNF algorithm scalability





Top-k algorithms results





Conclusion

- Proposed algorithms that use inverted indexes to efficiently search matching DNF/CNF BEs
- Proposed top-k algorithms for BEs based on relevance scores