

Лекция 1:

Търсене чрез булеви заявки от
ключови думи. Обратен индекс.

stoyan@lml.bas.bg

Оценяване на извлечените документи

- **Точност:** Процентният дял на релевантните по отношение на потребността документи спрямо всички извлечени.
- **Обхват:** Процентът на извлечените релевантни документи спрямо всички релевантни документи в колекцията.

Матрица на срещания ключова дума/документ

	Антоний и Клеопатра	Юлий Цезар	Бурята	Хамлет	Отело	Макбет
Антоний	1	1	0	0	0	1
Брут	1	1	0	1	0	0
Цезар	1	1	0	1	1	1
Калпурния	0	1	0	0	0	0
Клеопатра	1	0	0	0	0	0
милост	1	0	1	1	1	1
по-лош	1	0	1	1	1	0
...						

1, ако пиесата съдържа
съответната дума, 0 иначе.

Вектори на срещанията

На всяка ключова дума съпоставяме вектор от 0/1 с размер броя на документите, който отразява срещанията

- За да отговорим на заявката **Брут и Цезар** и не **Калпурния**, извършваме побитови операции със съответните вектори:

• 110100 &

• 110111 &

• ~010000 =
101111

• 100100

↑ Антоний и Клеопатра
↑ Хамлет

	Антоний и Клеопатра	Юлий Цезар	Бурята	Хамлет	Отело	Макбет
Антоний	1	1	0	0	0	1
Брут	1	1	0	1	0	0
Цезар	1	1	0	1	1	1
Калпурния	0 (1)	1 (0)	0 (1)	0 (1)	0 (1)	0 (1)
Клеопатра	1	0	0	0	0	0
милост	1	0	1	1	1	1
по-лош	1	0	1	1	1	0
...						

- **Проблем:** Представянето на матрицата на срещанията изисква много много памет
- **Наблюдение:** Броят на единиците е малък. Матрицата е много разрежена.
- **Решение:** Запаметяваме само позициите на единиците

Етапи на предварителна обработка на текста

- Разбиване текста на единици (**Tokenization**)
 - изрази като: 25-милиметров, полу-сято...
- Нормализация
 - пълни, малки букви, съкращения, акроними: С.У., текст...
- Преобразуване до основни форми (**Stemming**)
 - дунавски → дунав, вървейки → вървя
- Премахване на **stop** думи
 - съюзи, междуметия, предлози...

Технологии и инструменти за предварителна обработка на текст

- Най-често базирани на експертни правила и речници
- използване на регулярни изрази и релации
- използване на речници за съкращения, акроними, ...
- правила за преобразуване и замяна
- използване на крайни автомати и преобразуватели

Построяване на обратен индекс

Документ 1

Антоний във Египет не ще напусне
пиршества и спални за други битки.
Цезар трупа злато, отблъсквайки сърца

Документ 2

Цезар заслужавал е смъртта си,
Антоний ще обича Цезар мъртъв
не толкоз силно, колкото Брут жив

Дума	Документ
антоний	1
египет	1
напусне	1
пиршества	1
спални	1
други	1
битки	1
цезар	1
трупа	1
злато	1
отблъсквайки	1
сърца	1
цезар	2
заслужавал	2
смъртта	2
антоний	2
обича	2
цезар	2
мъртъв	2
толкоз	2
силно	2
колкото	2
брут	2
жив	2

Сортираме списъка първо по думите, после по документи.

Дума	Документ	Дума	Документ
антоний	1	антоний	1
египет	1	антоний	2
напусне	1	битки	1
пиршества	1	брут	2
спални	1	други	1
други	1	египет	1
битки	1	жив	2
цезар	1	заслужавал	2
трупа	1	злато	1
злато	1	колкото	2
отблъсквайки	1	мъртъв	2
сърца	1	напусне	1
цезар	2	обича	2
заслужавал	2	отблъсквайки	1
смъртта	2	пиршества	1
антоний	2	силно	2
обича	2	смъртта	2
цезар	2	спални	1
мъртъв	2	сърца	1
толкоз	2	толкоз	2
силно	2	трупа	1
колкото	2	цезар	1
брут	2	цезар	2
жив	2	цезар	2

Създаване на обратен индекс

- Обединяваме повторенията
- Разделяме речника от срещанията
- Добавяме поле за честота (броя) на срещанията

Дума	Документ
антоний	1
антоний	2
битки	1
брут	2
други	1
египет	1
жив	2
заслужавал	2
злато	1
колкото	2
мъртъв	2
напусне	1
обича	2
отблъсквайки	1
пиршества	1
силно	2
смъртта	2
спални	1
сърца	1
толкоз	2
трупа	1
цезар	1
цезар	2
цезар	2

Речник		Списък срещания	
Дума	Честота		
антоний	2	→ 1 → 2	
битки	1	→ 1	
брут	1	→ 2	
други	1	→ 1	
египет	1	→ 1	
жив	1	→ 2	
заслужавал	1	→ 2	
злато	1	→ 1	
колкото	1	→ 2	
мъртъв	1	→ 2	
напусне	1	→ 1	
обича	1	→ 2	
отблъсквайки	1	→ 1	
пиршества	1	→ 1	
силно	1	→ 2	
смъртта	1	→ 2	
спални	1	→ 1	
сърца	1	→ 1	
толкоз	1	→ 2	
трупа	1	→ 1	
цезар	2	→ 1 → 2	

повторение {

Булево търсене чрез обратен индекс

Реализиране на конюнкция: сливане на срещанията

Разгледаме заявката Брут и Цезар

ВАЖНО: Списъците трябва да са сортирани по номер на документ

Брут

2 → 4 → 8 → 16 → 32 → 64 → 128

Цезар

1 → 2 → 3 → 5 → 8 → 13 → 21 → 34

Алгоритъм за сливане на сортирани списъци

Алгоритъм за произволни булеви заявки

$\underbrace{(\text{Брут или Цезар})}_A$ и $\underbrace{(\text{Антоний или не Клеопатра})}_B$

$A \cup B \rightarrow C$

Сливаме всеки от подизразите и получаваме списък от срещания, след което сливаме получените списъци до получаването на срещанията за цялата заявка:

Конюнкция на n терма:

Оптимизация на изпълнението на заявката: започваме с най-късите списъци и сливаме в нарастващ ред

Позиционен обратен индекс

Съставни имена, фрази и изрази

Примери: Иван Иванов, Стара Загора, операционна система...

- Първо решение — добавяне на двойки от думи към речника
- Огромно нарастване на речника и списъците от срещания
- При нужда от намиране на тройки или четворки е необходимо разбиване на двойки и проверка дали са последователни
- Има смисъл само за често срещани съставни имена и изрази

Алтернативно решение: използване на позиционен индекс

Към списъка от срещания на всяко срещане на терм в документ добавяме списък на позициите в документа на съответните срещания

Пример: да бъде или да не бъде:

да: позиции

2	->	1, 17, 74, 222, 551;
4	->	8, 16, 190, 429, 433;
7	->	13, 23, 191;
...		

↑
документ

бъде:

1	->	17, 19;
4	->	17, 191, 291, 430, 434;
5	->	14, 19, 101;
...		

- Позволява в булевата заявка да се добави ограничение за разстоянието между термовете:
 - да и/1 бъде
 - компютърна и/3 мрежа
- Изисква съществено допълване на алгоритмите за сливане
- Изисква между 2 и 4 пъти повече памет за представяне на индексите.

Толериране на форми на дадена дума

- Чрез използване на stemming:
 - позиционен -> позиция, дунавската -> дунав, ...
 - втори речник върху “основите” на думите, за всяка основа се посочва списък от ключови думи със съответната основа
- Води до повече резултати, в някои случай - нерелевантни:
 - хлебарка -> хляб, националистически -> национален -> национал, ...
- Не се толерират грешки в изписването