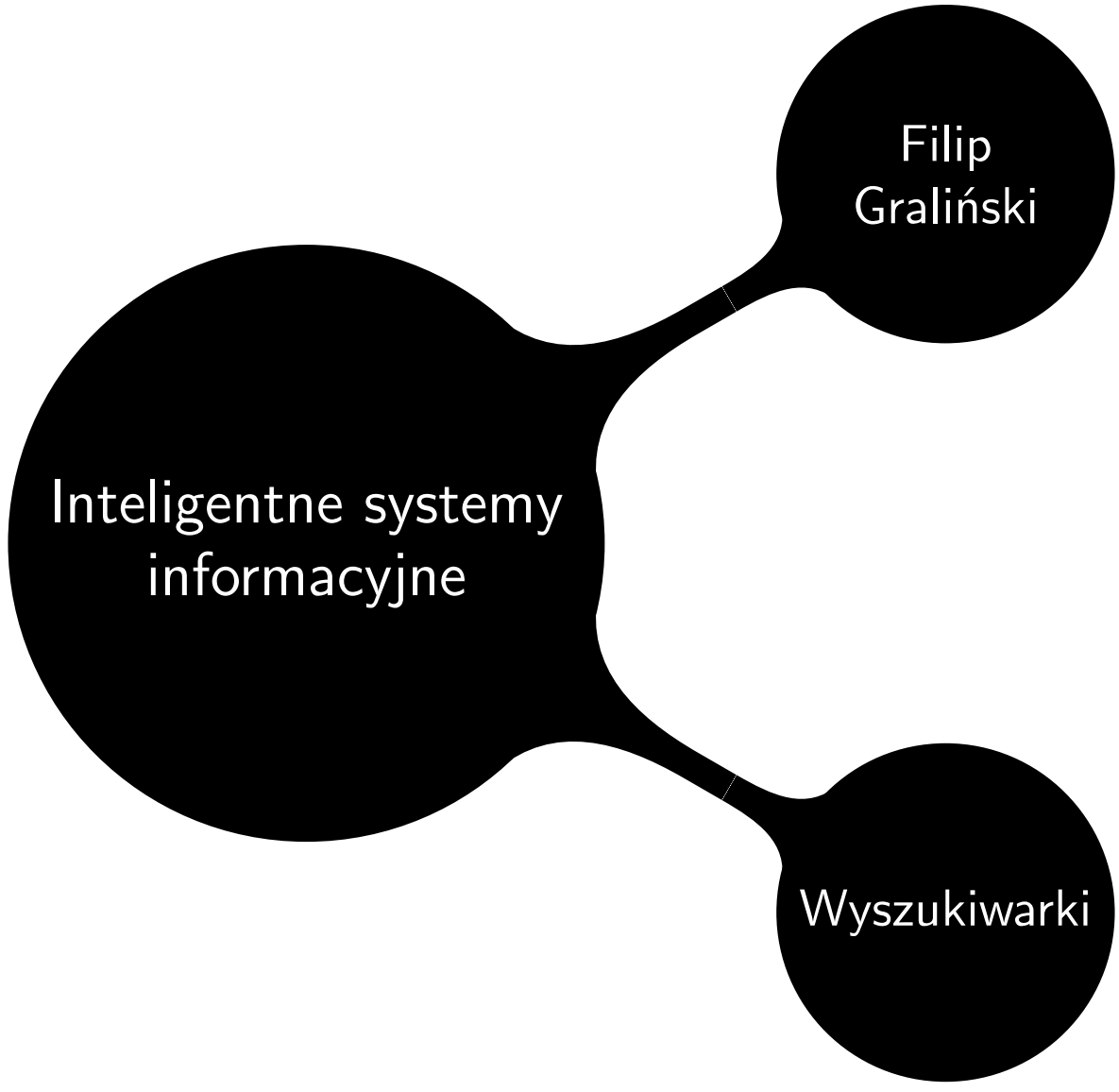
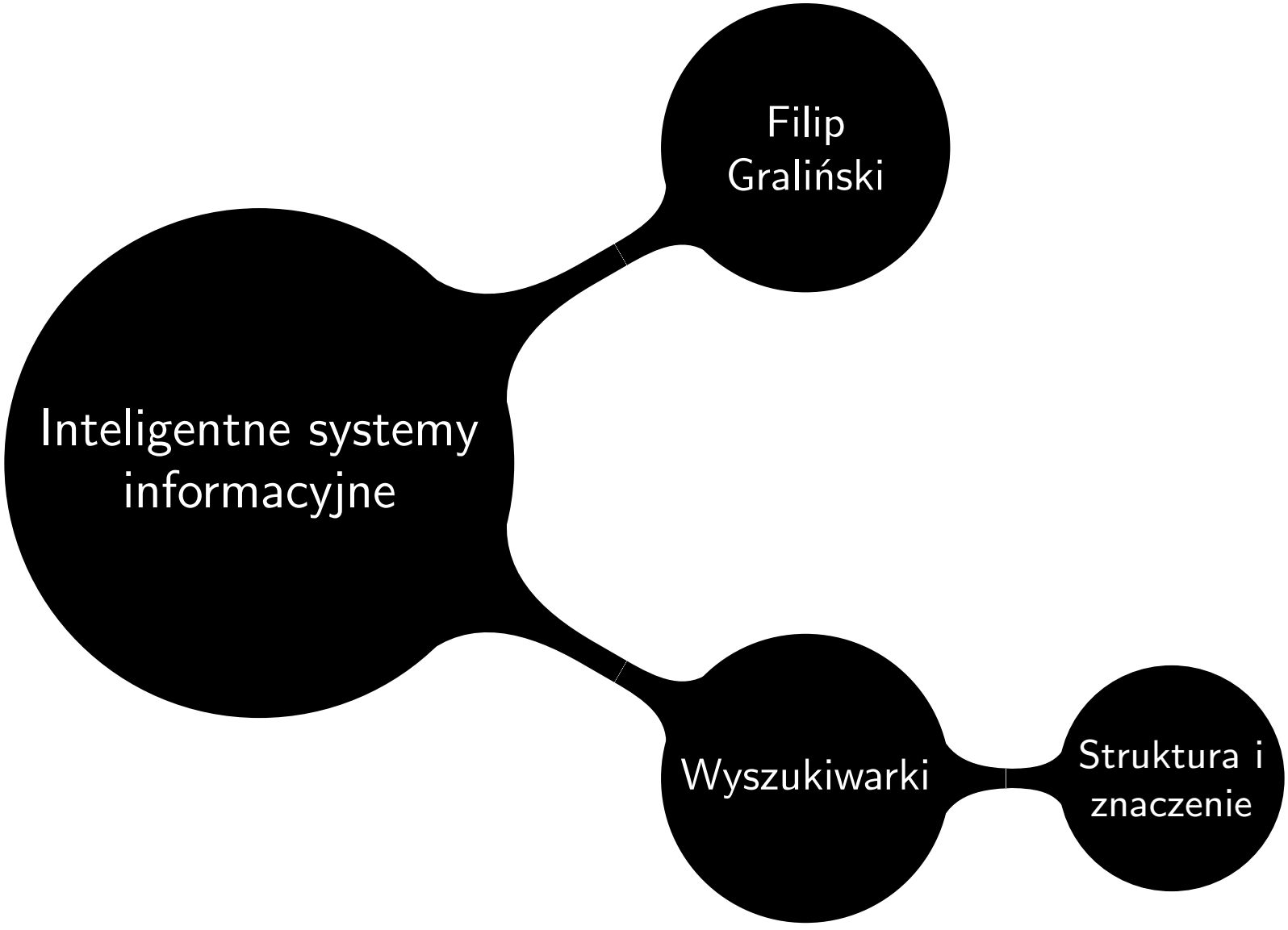
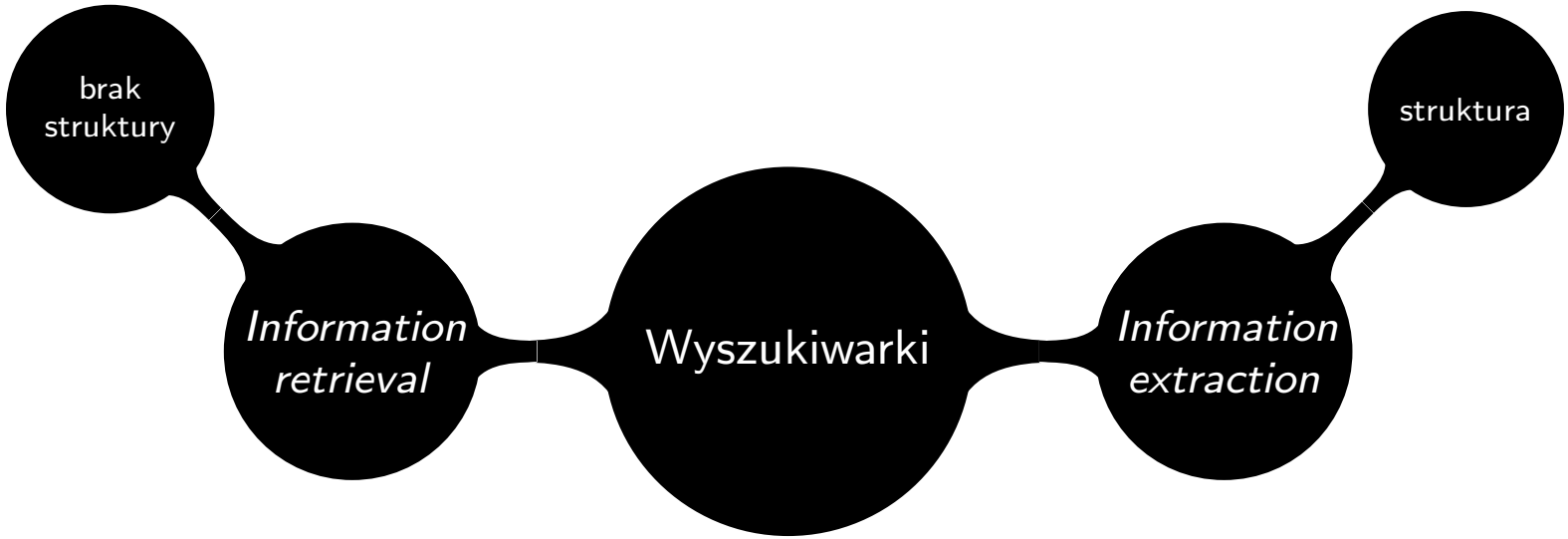
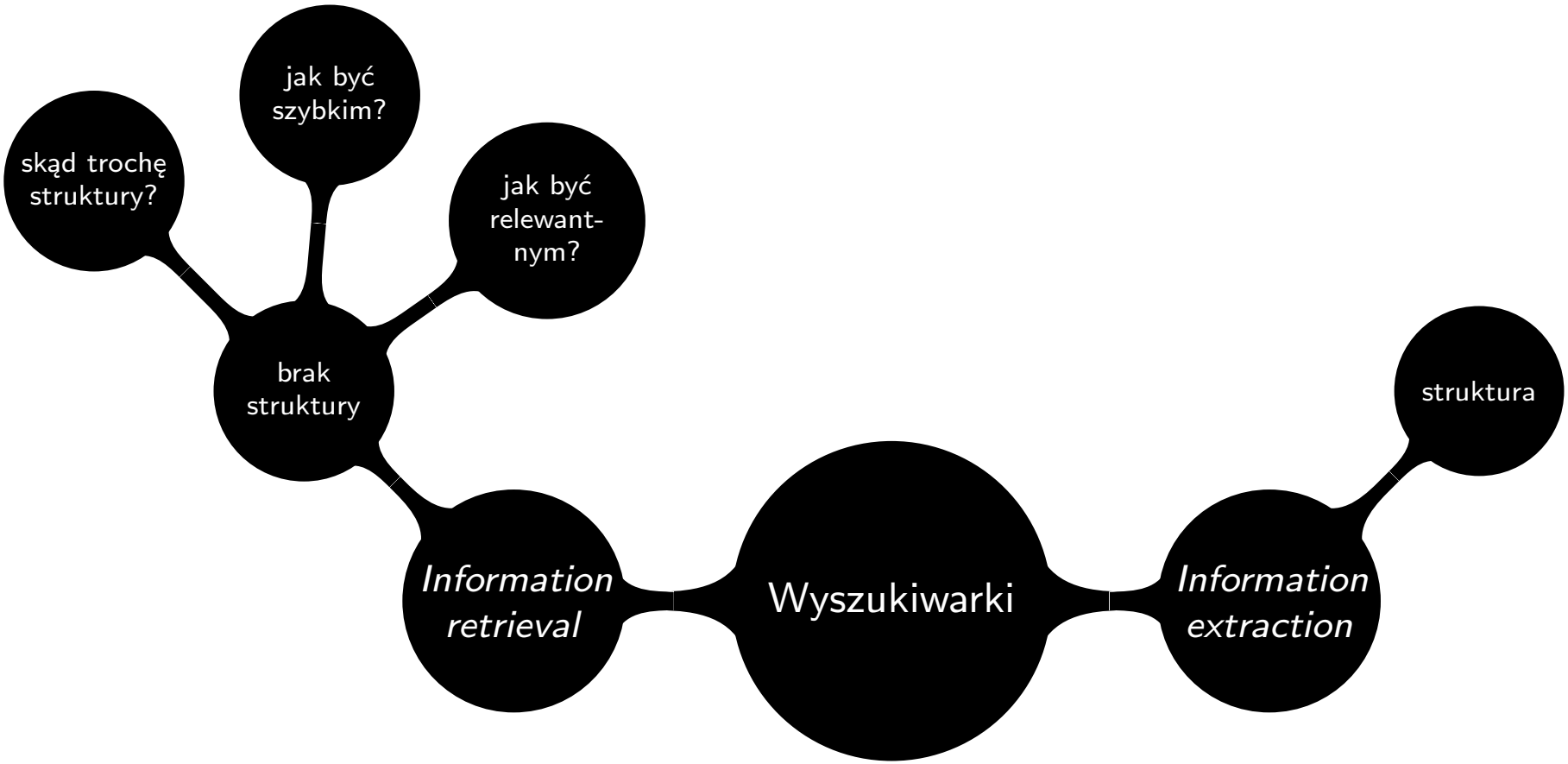
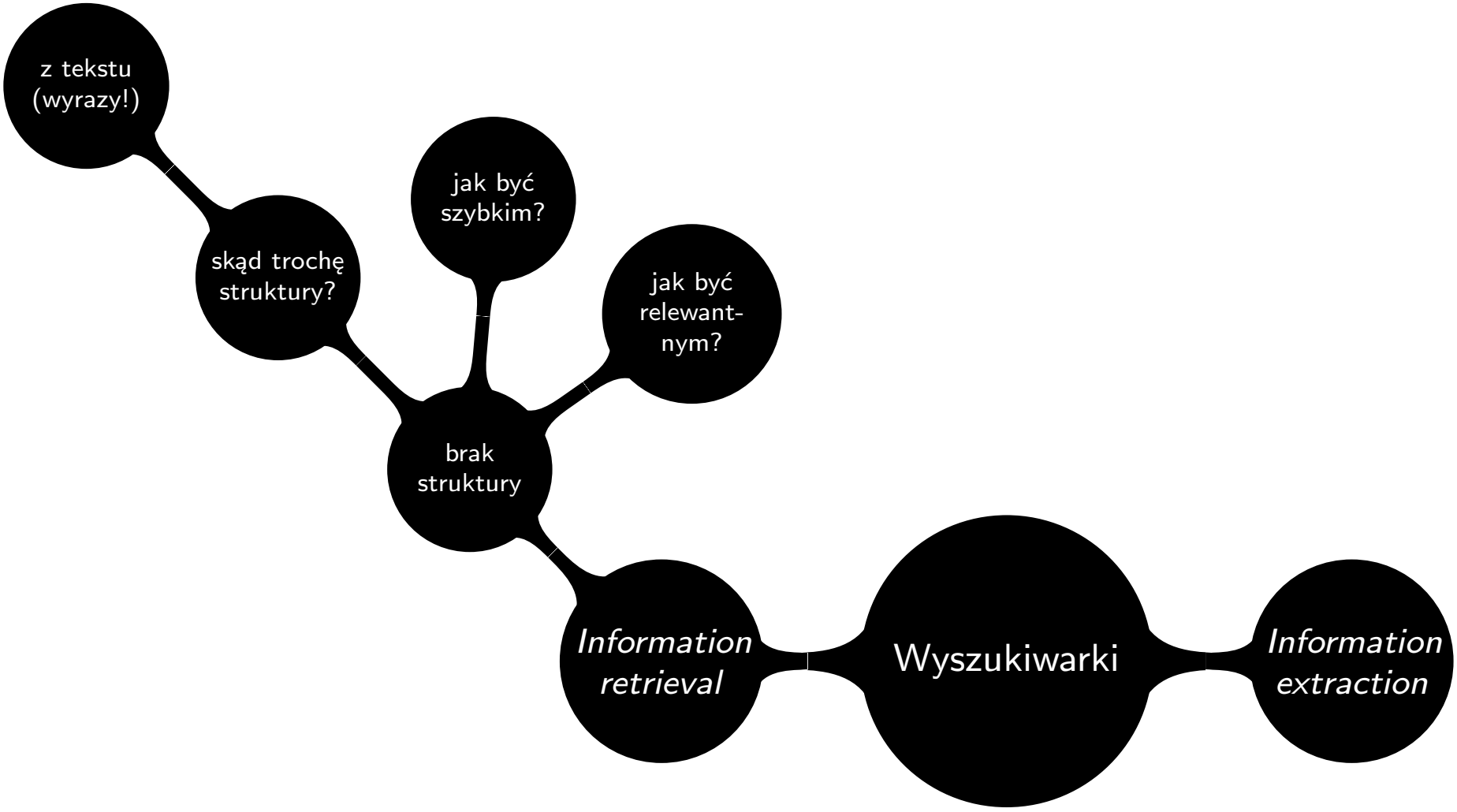

Filip
Graliński
Inteligentne systemy
informacyjne
Wyszukiwarki


Filip
Graliński
Inteligentne systemy
informacyjne
Wyszukiwarki
Struktura i
znaczenie

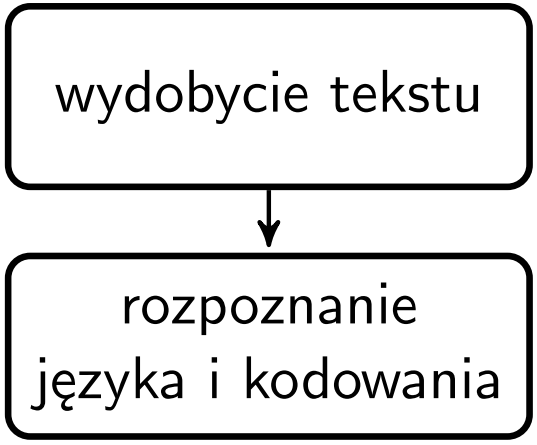

Information retrieval
Wyszukiwarki
Information extraction


brak
struktury
Information
retrieval
Wyszukiwarki
Information
extraction
struktura

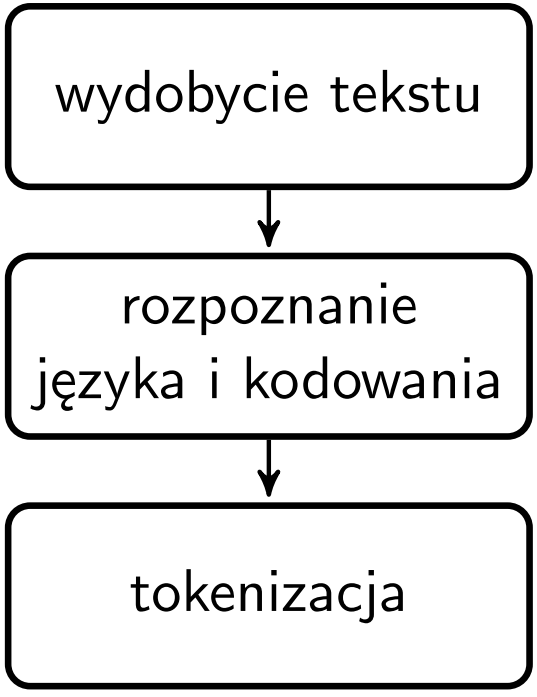

jak być szybkim?
skąd trochę struktury?
jak być relewant-nym?
brak struktury
Information retrieval
Wyszukiwarki
Information extraction
struktura


z tekstu
(wyrazy!)
skąd trochę
struktury?
jak być
szybkim?
jak być
relewant-
nym?
brak
struktury
Information
retrieval
Wyszukiwarki
Information
extraction

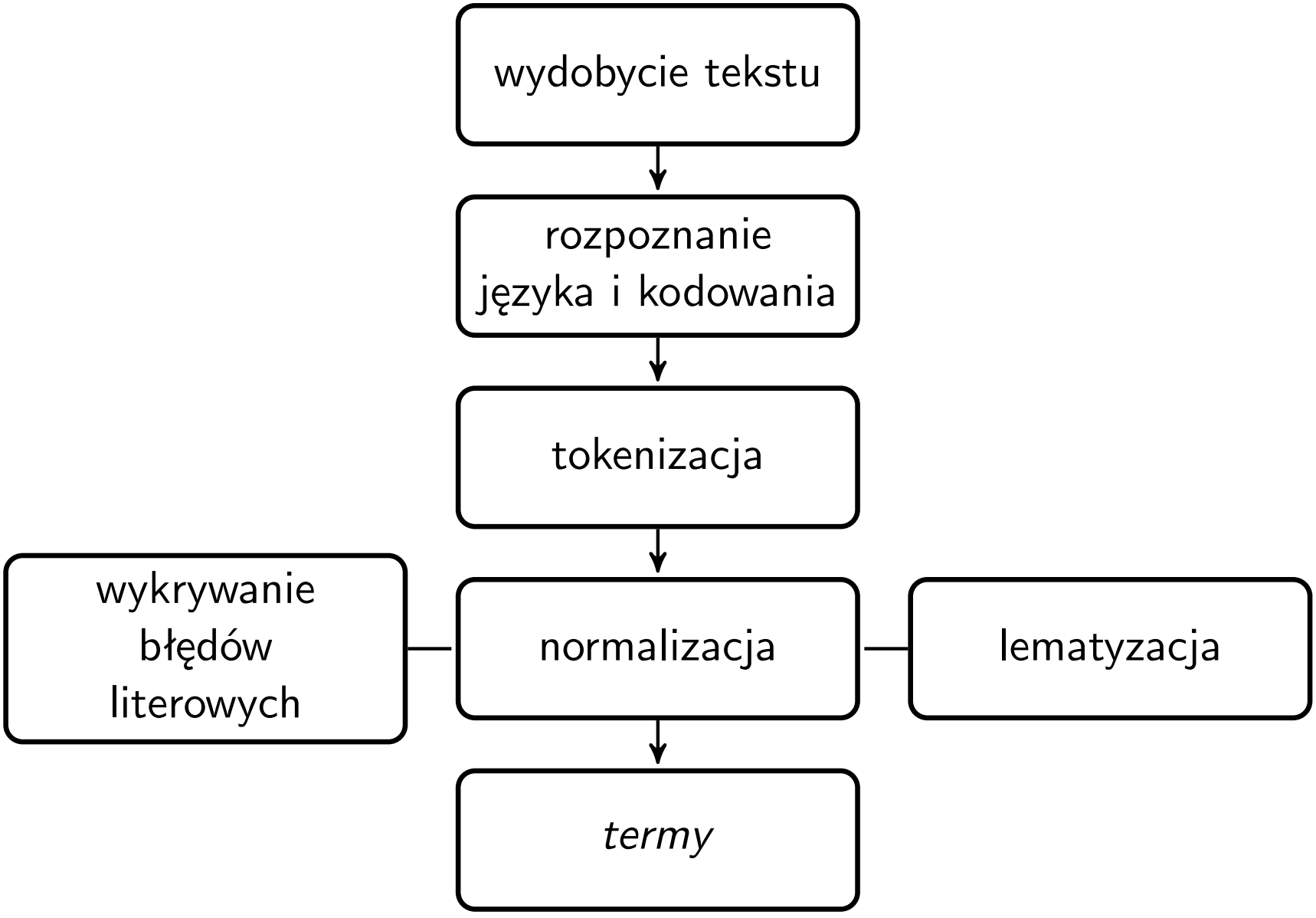
przetwarzanie dokumentu – od tekstu do *termów*



przetwarzanie dokumentu – od tekstu do *termów*



przetwarzanie dokumentu – od tekstu do *termów*



tokenizacja

Tokenizacja to podział na wyrazopodobne jednostki (tokeny)

- trudności dla języka angielskiego:

tokenizacja

Tokenizacja to podział na wyrazopodobne jednostki (tokeny)

- trudności dla języka angielskiego:
  - łącznik
  - apostrofy

tokenizacja

Tokenizacja to podział na wyrazopodobne jednostki (tokeny)

- trudności dla języka angielskiego:
  - łącznik
  - apostrofy
  - daty, numery telefonów, adresy e-mailowe etc.

tokenizacja

Tokenizacja to podział na wyrazopodobne jednostki (tokeny)

- trudności dla języka angielskiego:
  - łącznik
  - apostrofy
  - daty, numery telefonów, adresy e-mailowe etc.
  - tokeny typu *C++*, *C#*, *Yahoo!*

tokenizacja

Tokenizacja to podział na wyrazopodobne jednostki (tokeny)

- trudności dla języka angielskiego:
  - łącznik
  - apostrofy
  - daty, numery telefonów, adresy e-mailowe etc.
  - tokeny typu *C++*, *C#*, *Yahoo!*
- inne języki:
  - chiński – brak spacji

tokenizacja

Tokenizacja to podział na wyrazopodobne jednostki (tokeny)

- trudności dla języka angielskiego:
  - łącznik
  - apostrofy
  - daty, numery telefonów, adresy e-mailowe etc.
  - tokeny typu *C++*, *C#*, *Yahoo!*
- inne języki:
  - chiński – brak spacji
  - niemiecki – wyrazy złożone

tokenizacja

Tokenizacja to podział na wyrazopodobne jednostki (tokeny)

- trudności dla języka angielskiego:
  - łącznik
  - apostrofy
  - daty, numery telefonów, adresy e-mailowe etc.
  - tokeny typu *C++*, *C#*, *Yahoo!*
- inne języki:
  - chiński – brak spacji
  - niemiecki – wyrazy złożone
  - jęz. romańskie – zbitki typu *l'ordinateur*

Normalizacja

Czy powinny wyjść identyczne wyniki dla zapytań:

Uwaga! Spójność!

Zapytania powinny być normalizowane tak samo jak dokumenty

Normalizacja

Czy powinny wyjść identyczne wyniki dla zapytań:

- `Łódź` i `Łódź` ? *lowercasing*
- `koń` i `kon` ? usuwanie diakrytyków

Uwaga! Spójność!

Zapytania powinny być normalizowane tak samo jak dokumenty

Normalizacja

Czy powinny wyjść identyczne wyniki dla zapytań:

- `Łódź` i `Łódź` ? *lowercasing*
- `koń` i `kon` ? usuwanie diakrytyków
- `legenda` i `legendzie` ? lematyzacja

Uwaga! Spójność!

Zapytania powinny być normalizowane tak samo jak dokumenty

Normalizacja

Czy powinny wyjść identyczne wyniki dla zapytań:

- `Łódź` i `Łódź` ? *lowercasing*
- `koń` i `kon` ? usuwanie diakrytyków
- `legenda` i `legendzie` ? lematyzacja
- `urządzić` i `urządzenie` ?

Uwaga! Spójność!

Zapytania powinny być normalizowane tak samo jak dokumenty

Normalizacja

Czy powinny wyjść identyczne wyniki dla zapytań:

- `Łódź` i `Łódź` ? *lowercasing*
- `koń` i `kon` ? usuwanie diakrytyków
- `legenda` i `legendzie` ? lematyzacja
- `urządzić` i `urządzenie` ?
- `legenda` i `legendowy` ? rdzeniowanie

Uwaga! Spójność!

Zapytania powinny być normalizowane tak samo jak dokumenty

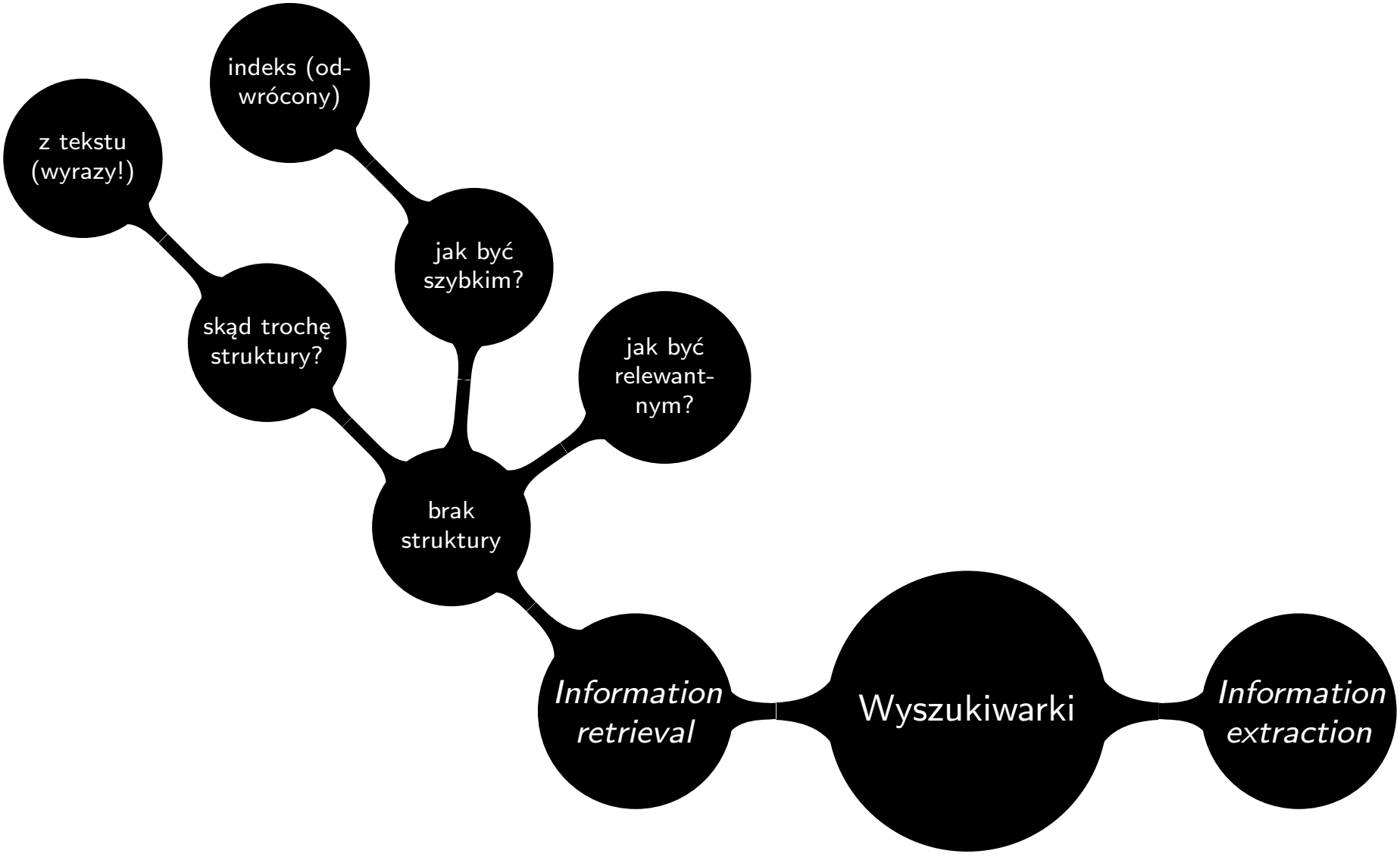
Normalizacja

Czy powinny wyjść identyczne wyniki dla zapytań:

- `łódź` i `Łódź` ? *lowercasing*
- `koń` i `kon` ? usuwanie diakrytyków
- `legenda` i `legendzie` ? lematyzacja
- `urządzić` i `urządzenie` ?
- `legenda` i `legendowy` ? rdzeniowanie
- `bałwan` i `bałwanek` ?

Uwaga! Spójność!

Zapytania powinny być normalizowane tak samo jak dokumenty


Wyszukiwarki
Information retrieval
Information extraction
brak struktury
skąd trochę struktury?
z tekstu (wyrazy!)
jak być szybkim?
indeks (od-wrócony)
jak być relewant-nym?

odwrócony indeks (*inverted index*)

doc1	*Ala ma kota.*
doc2	*Podobno jest kot w butach.*
doc3	*Ty chyba masz kota!*
doc4	*But chyba zgubiłem.*

odwrócony indeks (*inverted index*)

doc1	*Ala ma kota.*
doc2	*Podobno jest kot w butach.*
doc3	*Ty chyba masz kota!*
doc4	*But chyba zgubiłem.*

stop words: *jest*, *w*

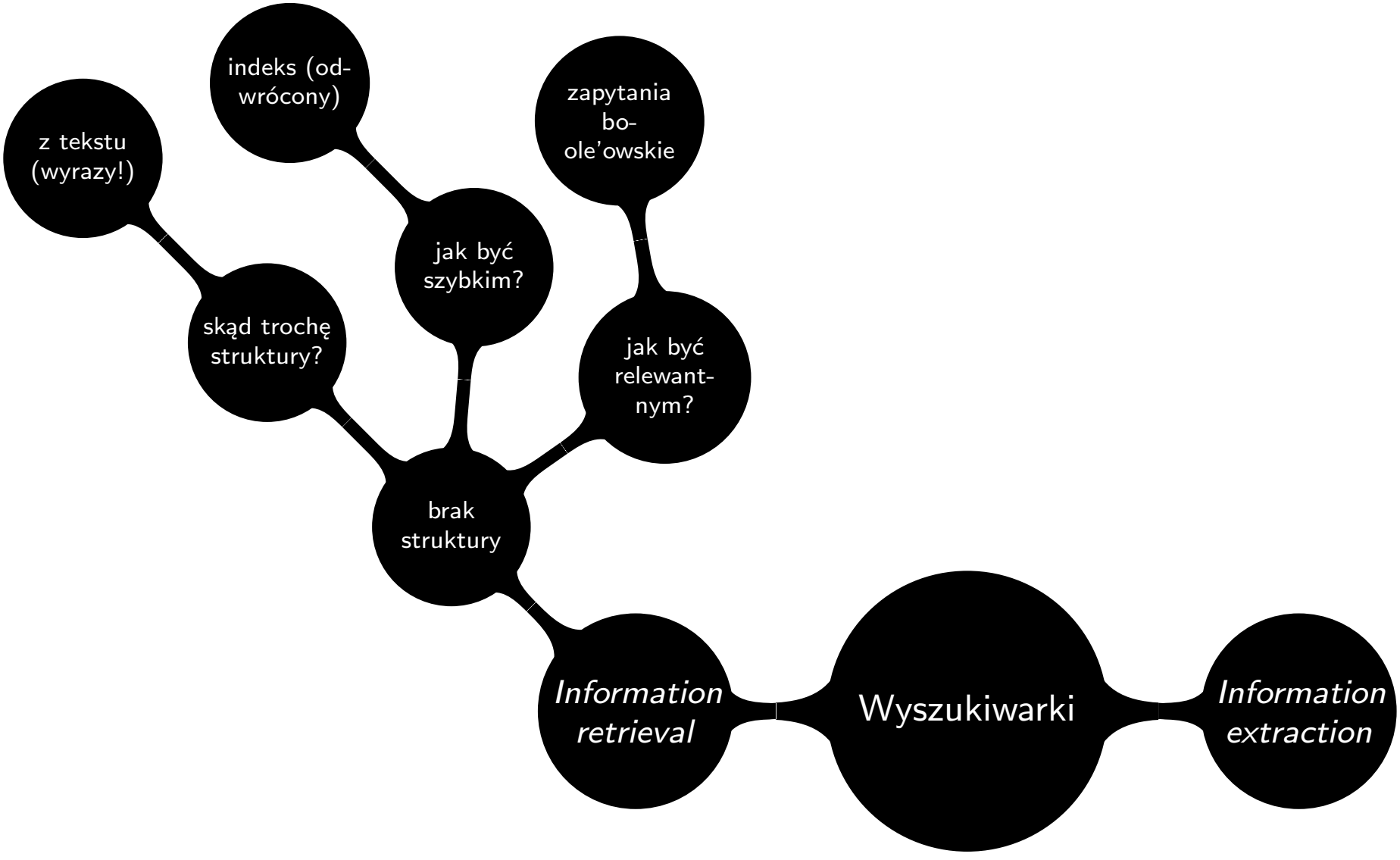
odwrócony indeks (*inverted index*)

doc1	*Ala ma kota.*
doc2	*Podobno jest kot w butach.*
doc3	*Ty chyba masz kota!*
doc4	*But chyba zgubiłem.*

stop words: *jest*, *w*

ala	$\rightarrow$	1		
but	$\rightarrow$	2	4	
chyba	$\rightarrow$	3	4	
kot	$\rightarrow$	1	2	3
mieć	$\rightarrow$	?		
podobno	$\rightarrow$	2		
ty	$\rightarrow$	3		
zgubić	$\rightarrow$	4		

Lista wszystkich termów w kolekcji to słownik (*vocabulary*):
$\{$*ala*, *but*, *chyba*, *kot*, *mieć*, *podobno*, *ty*, *zgubić*$\}$


Wyszukiwarki
Information retrieval
Information extraction
brak struktury
jak być szybkim?
jak być relewantnym?
skąd trochę struktury?
indeks (odwrócony)
zapytania boole'owskie
z tekstu (wyrazy!)

zapytania boole'owskie

- *pizzeria Poznań dowóz*

zapytania boole'owskie

- *pizzeria* AND *Poznań* AND *dowóz*

zapytania boole'owskie

- *pizzeria* AND *Poznań* AND *dowóz*
- (*pizzeria* OR *pizza* OR *tratoria*) AND *Poznań* AND *dowóz*

zapytania boole'owskie

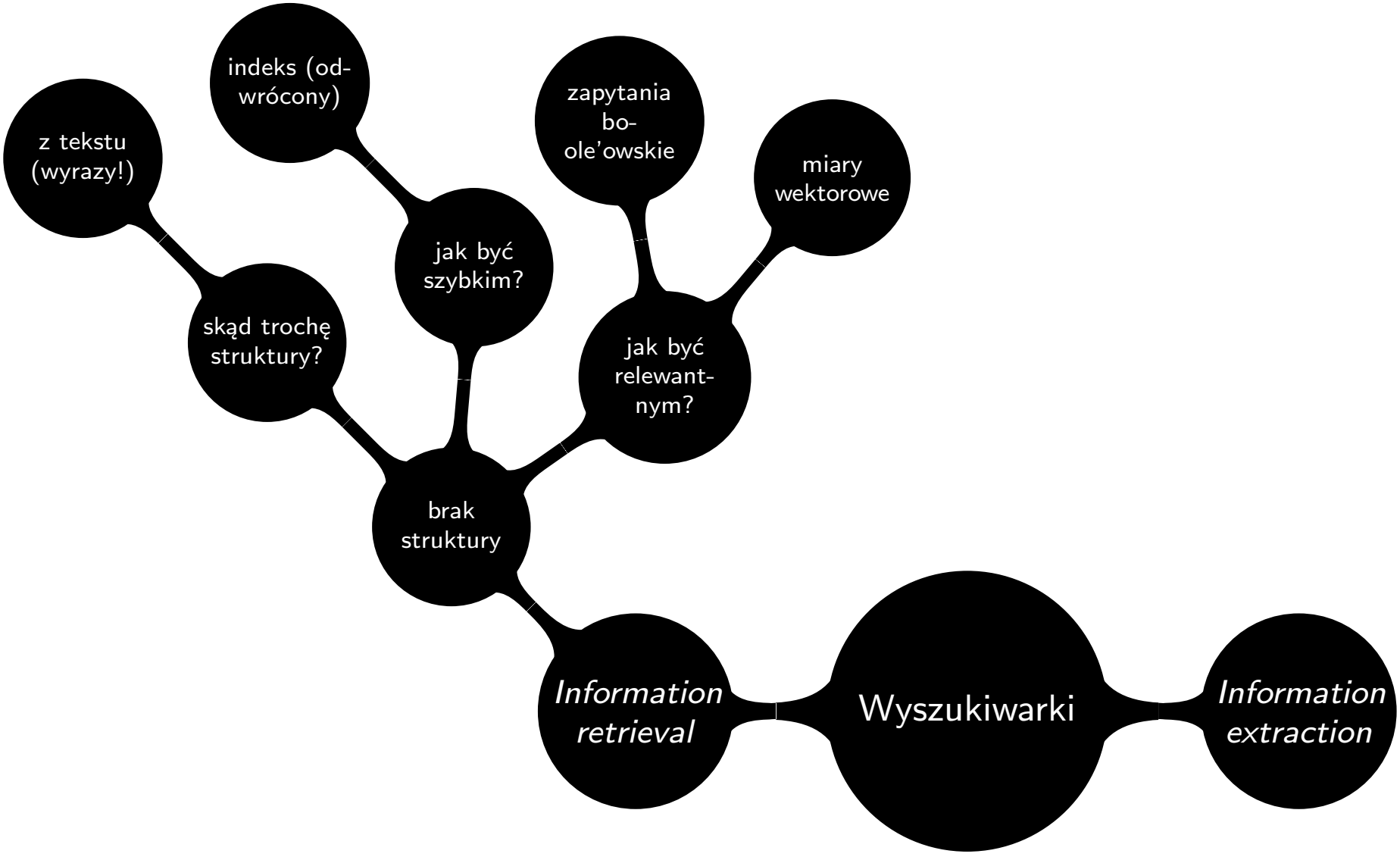
- *pizzeria* AND *Poznań* AND *dowóz*
- (*pizzeria* OR *pizza* OR *tratoria*) AND *Poznań* AND *dowóz*
- *pizzeria* AND *Poznań* AND *dowóz* AND NOT *golonka*

zapytania boole'owskie

- *pizzeria* AND *Poznań* AND *dowóz*
- (*pizzeria* OR *pizza* OR *tratoria*) AND *Poznań* AND *dowóz*
- *pizzeria* AND *Poznań* AND *dowóz* AND NOT *golonka*

Jak domyślnie interpretować zapytanie?

- jako zapytanie AND – być może za mało dokumentów
- rozwiązanie pośrednie?
- jako zapytanie OR – być może za dużo dokumentów


Wyszukiwarki
Information retrieval
Information extraction
brak struktury
skąd trochę struktury?
jak być szybkim?
jak być relewant-nym?
z tekstu (wyrazy!)
indeks (od-wrócony)
zapytania bo-ole'owskie
miary wektorowe

dokument jako wektor

doc1	*Ala ma kota.*
doc2	*Podobno jest kot w butach.*
doc3	*Ty chyba masz kota!*
doc4	*But chyba zgubiłem.*

dokument jako wektor

doc1	*Ala ma kota.*
doc2	*Podobno jest kot w butach.*
doc3	*Ty chyba masz kota!*
doc4	*But chyba zgubiłem.*

	ala	*but*	*chyba*	*kot*	*mieć*	*podobno*	*ty*	*zgubić*
doc1	1	0	0	1	1	0	0	0
doc2	0	1	0	1	0	1	0	0
doc3	0	0	1	1	1	0	1	0
doc4	0	1	1	0	0	0	0	1

dokument jako wektor

doc1	*Ala ma kota.*
doc2	*Podobno jest kot w butach.*
doc3	*Ty chyba masz kota!*
doc4	*But chyba zgubiłem.*
doc5	*Kot ma kota.*

	ala	*but*	*chyba*	*kot*	*mieć*	*podobno*	*ty*	*zgubić*
doc1	1	0	0	1	1	0	0	0
doc2	0	1	0	1	0	1	0	0
doc3	0	0	1	1	1	0	1	0
doc4	0	1	1	0	0	0	0	1

dokument jako wektor

doc1	*Ala ma kota.*
doc2	*Podobno jest kot w butach.*
doc3	*Ty chyba masz kota!*
doc4	*But chyba zgubiłem.*
doc5	*Kot ma kota.*

	ala	*but*	*chyba*	*kot*	*mieć*	*podobno*	*ty*	*zgubić*
doc1	1	0	0	1	1	0	0	0
doc2	0	1	0	1	0	1	0	0
doc3	0	0	1	1	1	0	1	0
doc4	0	1	1	0	0	0	0	1
doc5	0	0	0	?	1	0	0	0

jak uwzględnić frekwencję wyrazu?

$$\mathrm{tf}_{t,d}$$

jak uwzględnić frekwencję wyrazu?

$$\mathrm{tf}_{t,d}$$

$$1 + \log(\mathrm{tf}_{t,d})$$

jak uwzględnić frekwencję wyrazu?

$$\mathsf{tf}_{t,d}$$

$$1 + \log(\mathsf{tf}_{t,d})$$

$$\begin{cases} 1, & \text{jeśli } \mathsf{tf}_{t,d} > 0 \\ 0, & \text{w przeciwnym razie} \end{cases}$$

jak uwzględnić frekwencję wyrazu?

$$\mathrm{tf}_{t,d}$$

$$1 + \log(\mathrm{tf}_{t,d})$$

$$\begin{cases} 1, & \text{jeśli } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{w przeciwnym razie} \end{cases}$$

$$0.5 + \frac{0.5 \times tf_{t,d}}{max_t(\mathrm{tf}_{t,d})}$$

odwrotna częstość w dokumentach

Czy wszystkie wyrazy są tak samo ważne?

odwrotna częstość w dokumentach

Czy wszystkie wyrazy są tak samo ważne?

NIE. Wyrazy pojawiające się w wielu dokumentach są mniej ważne.

odwrotna częstość w dokumentach

Czy wszystkie wyrazy są tak samo ważne?

NIE. Wyrazy pojawiające się w wielu dokumentach są mniej ważne.

Aby to uwzględnić, przemnażamy frekwencję wyrazu przez odwrotną częstość w dokumentach (*inverse document frequency*):

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}$$

,

idf_t – odwrotna częstość wyrazu t w dokumentach

N – liczba dokumentów w kolekcji

df_f – w ilu dokumentach wystąpił wyraz t?

odwrotna częstość w dokumentach (cd.)

wyraz wystąpił...	$\text{idf}_t = \ldots$
w 1 dokumencie	

odwrotna częstość w dokumentach (cd.)

wyraz wystąpił...	$\mathrm{idf}_t = \ldots$
w 1 dokumencie	$= \log N/1 = \log N$

odwrotna częstość w dokumentach (cd.)

wyraz wystąpił...	$\text{idf}_t = \ldots$
w 1 dokumencie	$= \log N/1 = \log N$
2 razy w kolekcji	$= \log N/2$ lub $\log N$

odwrotna częstość w dokumentach (cd.)

wyraz wystąpił...	$\mathrm{idf}_t = \ldots$
w 1 dokumencie	$= \log N/1 = \log N$
2 razy w kolekcji	$= \log N/2$ lub $\log N$
w połowie dokumentów	$= \log N/(N/2) = \log 2$

odwrotna częstość w dokumentach (cd.)

wyraz wystąpił...	$\mathrm{idf}_t = \ldots$
w 1 dokumencie	$= \log N/1 = \log N$
2 razy w kolekcji	$= \log N/2$ lub $\log N$
w połowie dokumentów	$= \log N/(N/2) = \log 2$
we wszystkich dokumentach	$= \log N/N = \log 1 = 0$

odwrotna częstość w dokumentach (cd.)

wyraz wystąpił...	$\text{idf}_t = \ldots$
w 1 dokumencie	$= \log N/1 = \log N$
2 razy w kolekcji	$= \log N/2$ lub $\log N$
w połowie dokumentów	$= \log N/(N/2) = \log 2$
we wszystkich dokumentach	$= \log N/N = \log 1 = 0$

Zamiast $\text{tf}_{t,d}$ będziemy w wektorach rozpatrywać wartości:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

odwrotna częstość w dokumentach (cd.)

wyraz wystąpił...	$\text{idf}_t = \ldots$
w 1 dokumencie	$= \log N/1 = \log N$
2 razy w kolekcji	$= \log N/2$ lub $\log N$
w połowie dokumentów	$= \log N/(N/2) = \log 2$
we wszystkich dokumentach	$= \log N/N = \log 1 = 0$

Zamiast $\text{tf}_{t,d}$ będziemy w wektorach rozpatrywać wartości:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

Overlap score measure:

$$\text{Score}(q, d) = \sum_{t \in q} \text{tf-idf}_{t,d}$$