Similarity Thesauri and Cross-Language Retrieval

Sebastian Marius Kirsch
skirsch@moebius.inka.de

What is a thesaurus?

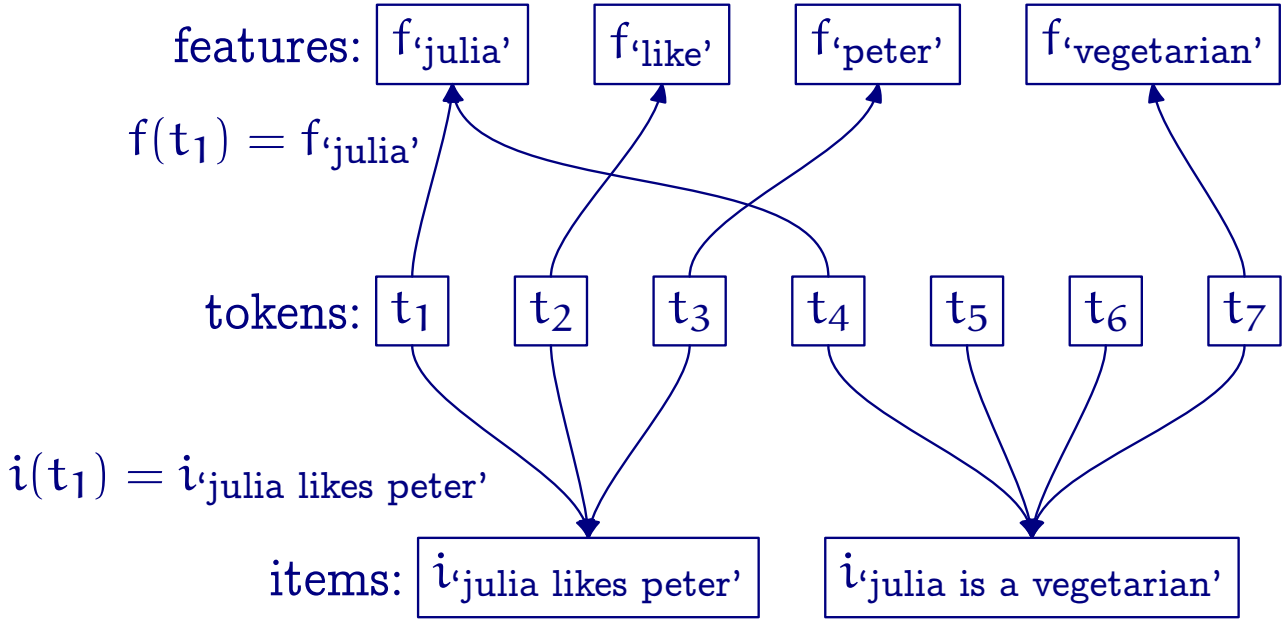
the·sau·rus [...]

2a: a book of words or of information about a particular field or set of concepts; *especially:* a book of words and their synonyms **b**: a list of subject headings or descriptors usually with a cross-reference system for use in the organization of a collection of documents for reference and retrieval

– definition from the Merriam-Webster Online Dictionary

Vector Space Model

Tokens, Items, and Features



Weighting

Weighting

$$\mathbf{ff}(f_i, i_j) = |\{\theta \in \Theta | f(\theta) = f_i \wedge i(\theta) = i_j\}|$$

Weighting

$$\mathrm{ff}(f_i, i_j) = |\{\theta \in \Theta | f(\theta) = f_i \wedge i(\theta) = i_j\}|$$

$$\mathrm{weight}_{\mathrm{ff}}(i_j, f_i) = \mathrm{ff}(f_i, i_j)$$

Weighting

$$\mathrm{ff}(f_i, i_j) = |\{\theta \in \Theta | f(\theta) = f_i \wedge i(\theta) = i_j\}|$$

$$\mathrm{weight}_{\mathrm{ff}}(i_j, f_i) = \mathrm{ff}(f_i, i_j)$$

$$\mathrm{if}(f_i) = |\{i \in I | \exists\, \theta \in \Theta : f(\theta) = f_i \wedge i(\theta) = i\}|$$

Weighting

$$\mathrm{ff}(f_i, i_j) = |\{\theta \in \Theta | f(\theta) = f_i \wedge i(\theta) = i_j\}|$$

$$\mathrm{weight}_{\mathrm{ff}}(i_j, f_i) = \mathrm{ff}(f_i, i_j)$$

$$\mathrm{if}(f_i) = |\{i \in I | \exists\, \theta \in \Theta : f(\theta) = f_i \wedge i(\theta) = i\}|$$

$$\mathrm{iif}(f_i) = \frac{1}{\log(\mathrm{if}(f_i) + 1)}$$

Weighting

$$\mathrm{ff}(f_i, i_j) = |\{\theta \in \Theta | f(\theta) = f_i \wedge i(\theta) = i_j\}|$$

$$\mathrm{weight}_{\mathrm{ff}}(i_j, f_i) = \mathrm{ff}(f_i, i_j)$$

$$\mathrm{if}(f_i) = |\{i \in I | \exists\, \theta \in \Theta : f(\theta) = f_i \wedge i(\theta) = i\}|$$

$$\mathrm{iif}(f_i) = \frac{1}{\log\left(\mathrm{if}(f_i) + 1\right)}$$

$$\mathrm{weight}_{\mathrm{tf}\cdot\mathrm{idf}}(i_j, f_i) = \mathrm{ff}(f_i, i_j) \cdot \mathrm{iif}(f_i)$$

Construction

- normalization: $\text{weight}_{\text{norm.}}(i_j, f_i) = \frac{\text{weight}(i_j, f_i)}{\sqrt{\sum_{f \in F} (\text{weight}(i_j, f))^2}}$
- split similarity computation

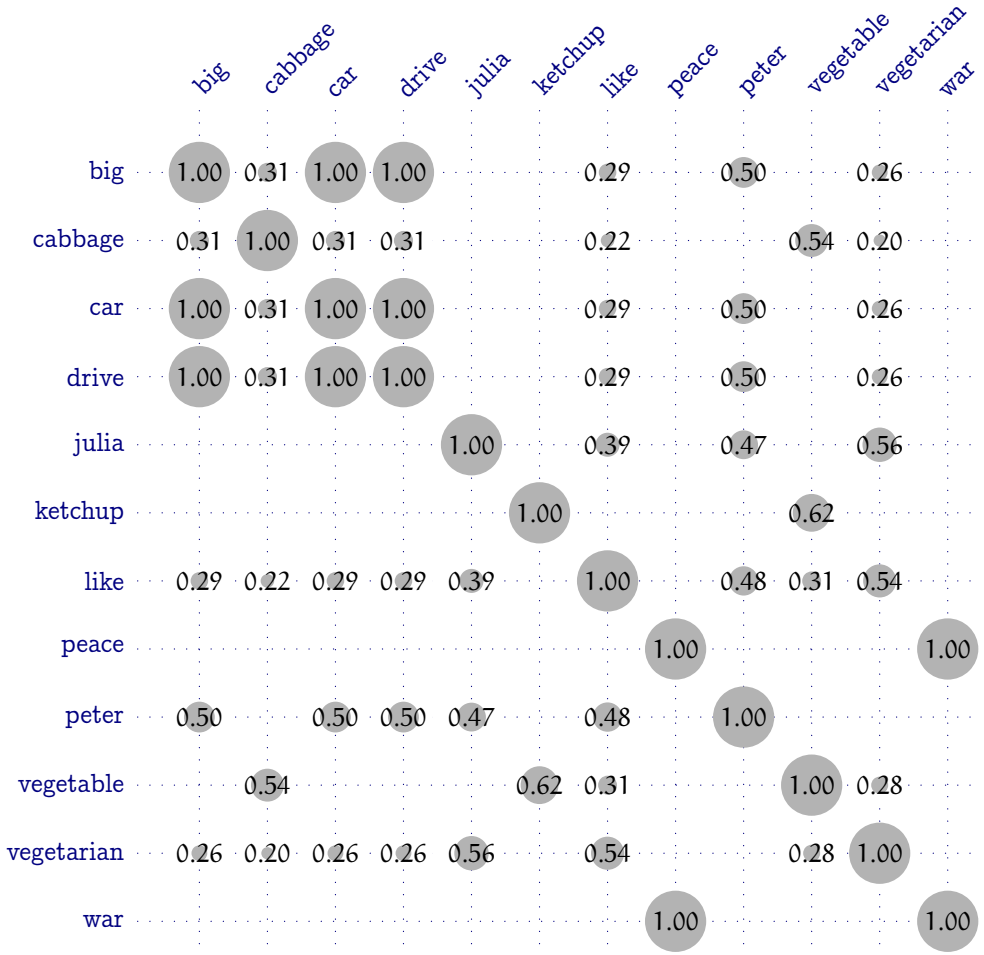
$$\text{SIM}(t_i, t_j) = \vec{t}_i \cdot \vec{t}_j$$

$$= \frac{\overbrace{\sum_{d \in D} \text{weight}(t_i, d) \cdot \text{weight}(t_j, d)}^{=\text{sim}(t_i, t_j)}}{\sqrt{\underbrace{\left(\sum_{d \in D} (\text{weight}(t_i, d))^2\right)}_{=c(t_i)} \cdot \underbrace{\left(\sum_{d \in D} (\text{weight}(t_i, d))^2\right)}_{=c(t_i)}}}$$

Monolingual thesaurus

Items big, cabbage, car, drive, julia, ketchup, like, peace, peter, vegetable, vegetarian, war

Features peter drives a big car, julia likes peter, julia is a vegetarian, vegetarians like vegetables, cabbage is a vegetable, big vegetarians who like cabbage do not drive cars, war is peace, ketchup is a vegetable


big
cabbage
car
drive
julia
ketchup
like
peace
peter
vegetable
vegetarian
war
big 1.00 0.31 1.00 1.00 0.29 0.50 0.26
cabbage 0.31 1.00 0.31 0.31 0.22 0.54 0.20
car 1.00 0.31 1.00 1.00 0.29 0.50 0.26
drive 1.00 0.31 1.00 1.00 0.29 0.50 0.26
julia 1.00 0.39 0.47 0.56
ketchup 1.00 0.62
like 0.29 0.22 0.29 0.29 0.39 1.00 0.48 0.31 0.54
peace 1.00 1.00
peter 0.50 0.50 0.50 0.47 0.48 1.00
vegetable 0.54 0.62 0.31 1.00 0.28
vegetarian 0.26 0.20 0.26 0.26 0.56 0.54 0.28 1.00
war 1.00 1.00

Back
Close

Query Expansion

Query Expansion

$$\vec{q}_c = \sum_{t \in T} \text{weight}(q, t) \cdot \vec{t}$$

Query Expansion

$$\vec{q}_c = \sum_{t \in T} \text{weight}(q, t) \cdot \vec{t}$$

$$\text{simqt}(q, t_i) := \vec{q}_c \cdot \vec{t}_i = \sum_{t \in T} \text{weight}(q, t) \cdot \vec{t} \cdot \vec{t}_i$$

Query Expansion

$$\vec{q}_c = \sum_{t \in T} \text{weight}(q, t) \cdot \vec{t}$$

$$\text{simqt}(q, t_i) := \vec{q}_c \cdot \vec{t}_i = \sum_{t \in T} \text{weight}(q, t) \cdot \vec{t} \cdot \vec{t}_i$$

$$\vec{q}_e = \left(\frac{\text{simqt}(q, t_1)}{\sum_{t \in T} \text{weight}(q, t)}, \ldots, \frac{\text{simqt}(q, t_n)}{\sum_{t \in T} \text{weight}(q, t)}\right)^\top$$

Query Expansion

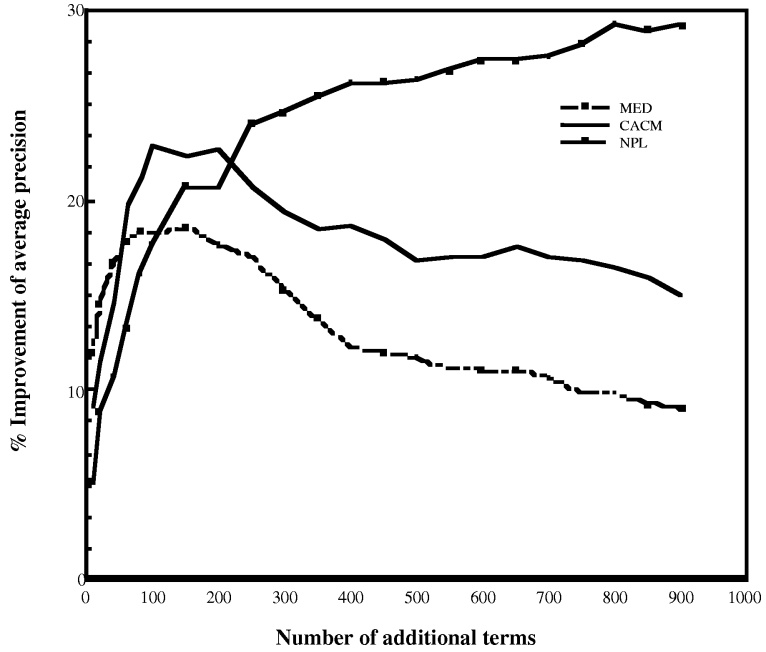
$$\vec{q}_c = \sum_{t \in T} \text{weight}(q, t) \cdot \vec{t}$$

$$\text{simqt}(q, t_i) := \vec{q}_c \cdot \vec{t}_i = \sum_{t \in T} \text{weight}(q, t) \cdot \vec{t} \cdot \vec{t}_i$$

$$\vec{q}_e = \left(\frac{\text{simqt}(q, t_1)}{\sum_{t \in T} \text{weight}(q, t)}, \ldots, \frac{\text{simqt}(q, t_n)}{\sum_{t \in T} \text{weight}(q, t)}\right)^\top$$

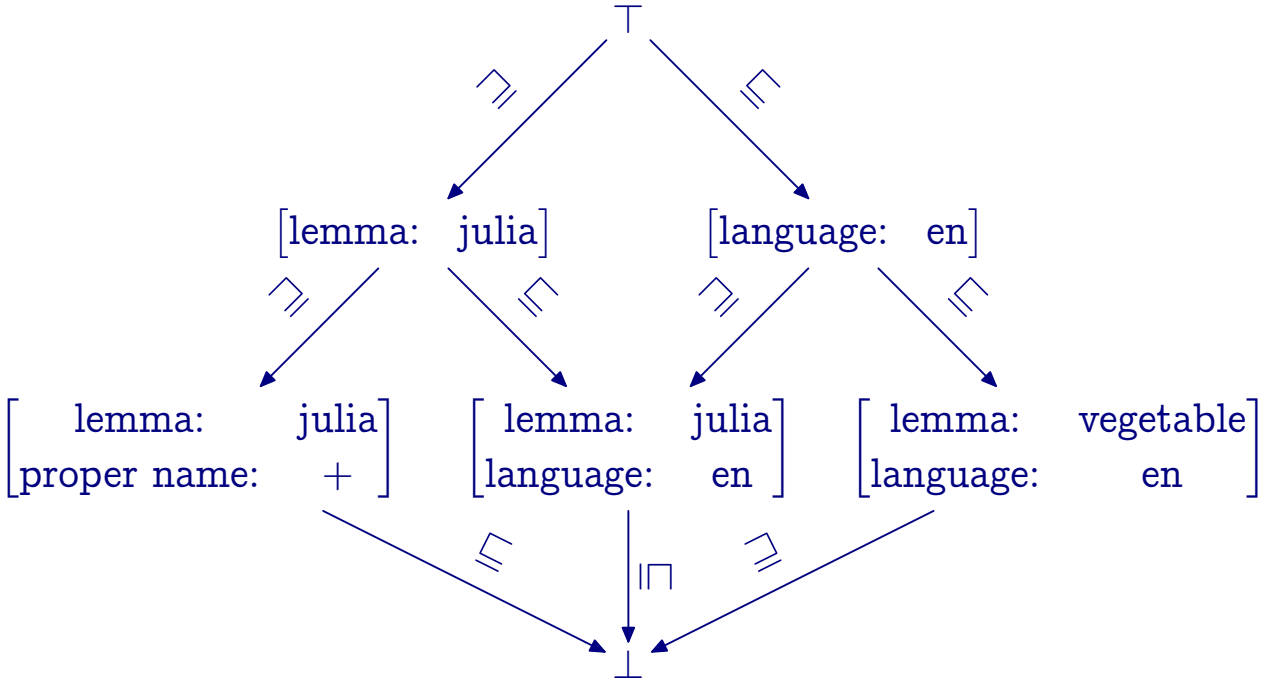
$$\vec{q}_{\text{exp.}} = \vec{q} + \vec{q}_e$$

Evaluation: Query Expansion



Cross-Language Information Retrieval

Subsumption order



Redefining ff, if

A feature f_i is deemed to occur in an item i_j for every token θ where

1. f_i is equal to or more general than that token's feature $f(\theta)$, and
2. i_j is equal to or more general than that token's item $i(\theta)$.

Redefining ff, if

A feature f_i is deemed to occur in an item i_j for every token θ where

1. f_i is equal to or more general than that token's feature $f(\theta)$, and
2. i_j is equal to or more general than that token's item $i(\theta)$.

$$\mathrm{ff}(f_i, i_j) = |\{\theta \in \Theta | f_i \sqsubseteq f(\theta) \wedge i_j \sqsubseteq i(\theta)\}|$$

Redefining ff, if

A feature f_i is deemed to occur in an item i_j for every token θ where

1. f_i is equal to or more general than that token's feature $f(\theta)$, and
2. i_j is equal to or more general than that token's item $i(\theta)$.

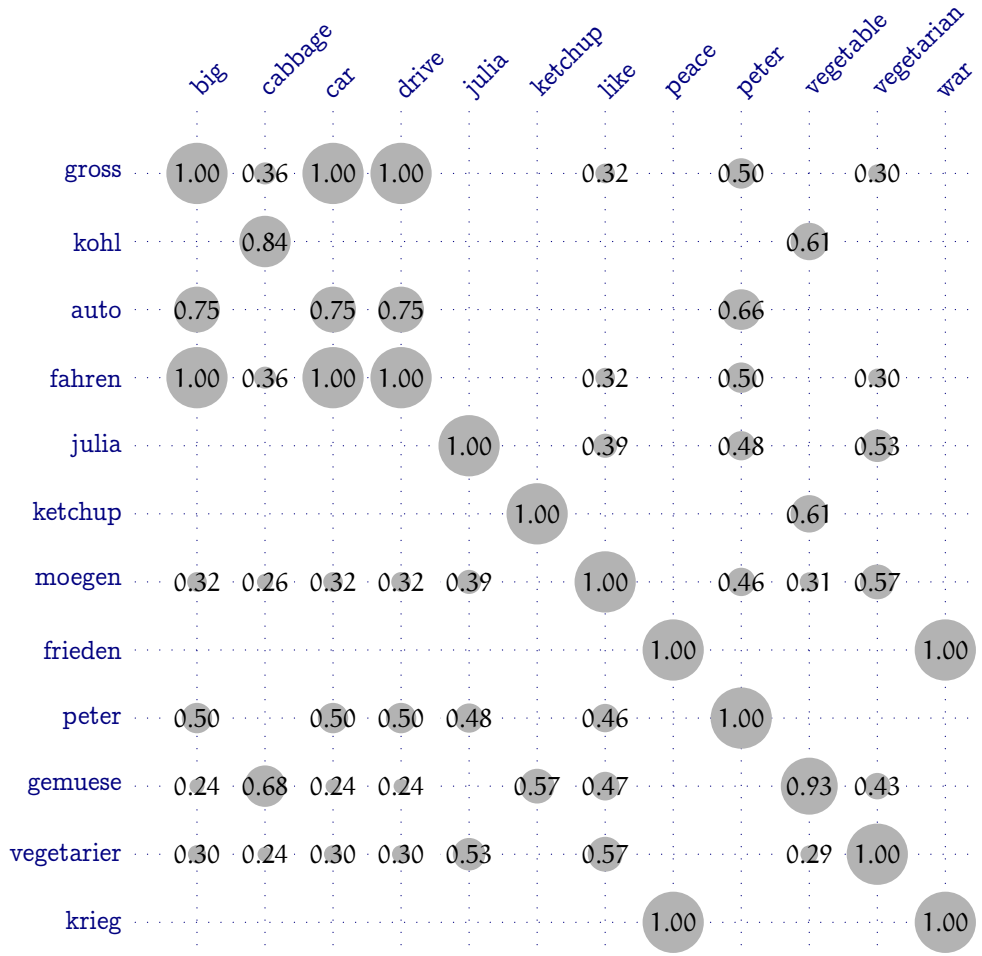
$$\mathrm{ff}(f_i, i_j) = |\{\theta \in \Theta | f_i \sqsubseteq f(\theta) \wedge i_j \sqsubseteq i(\theta)\}|$$

$$\mathrm{if}(f_i) = |\{i \in I | \exists\, \theta \in \Theta : f_i \sqsubseteq f(\theta) \wedge i_j \sqsubseteq i(\theta)\}|$$

Cross-lingual thesaurus

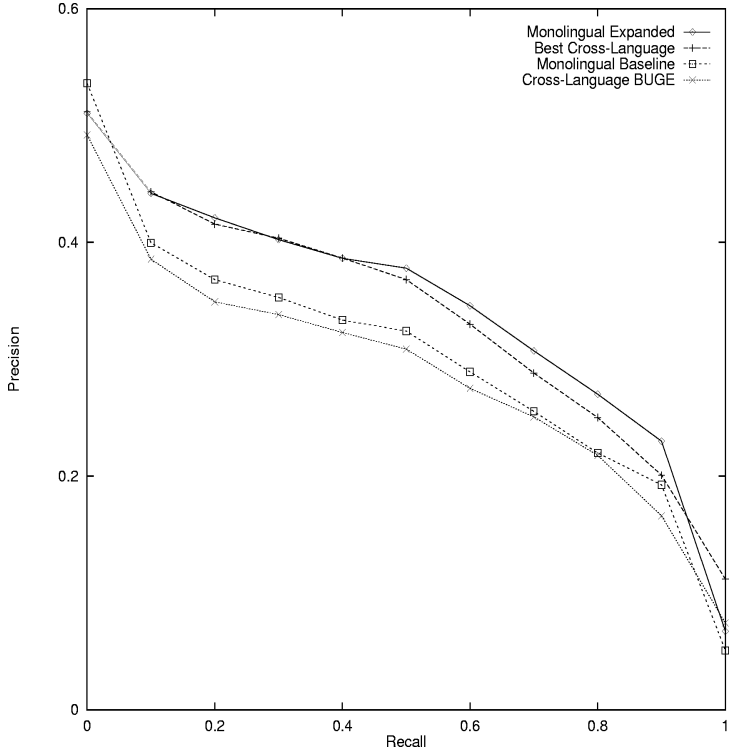
Items big, cabbage, car, drive, julia, ketchup, like, peace, peter, vegetable, vegetarian, war, gross, kohl, auto, fahren, julia, ketchup, moegen, frieden, peter, gemuese, vegetarier, krieg

Features [peter drives a big car, peter faehrt ein grosses auto], [julia likes peter, julia mag peter], [julia is a vegetarian, julia ist vegetarierin], [vegetarians like vegetables, vegetarier moegen gemuese], [cabbage is a vegetable, kohl ist gemuese], [big vegetarians who like cabbage do not drive cars, grosse vegetarier die gemuese moegen fahren keine autos], [war is peace, krieg ist frieden], [ketchup is a vegetable, ketchup ist gemuese]


big
cabbage
car
drive
julia
ketchup
like
peace
peter
vegetable
vegetarian
war
gross 1.00 0.36 1.00 1.00 0.32 0.50 0.30
kohl 0.84 0.61
auto 0.75 0.75 0.75 0.66
fahren 1.00 0.36 1.00 1.00 0.32 0.50 0.30
julia 1.00 0.39 0.48 0.53
ketchup 1.00 0.61
moegen 0.32 0.26 0.32 0.32 0.39 1.00 0.46 0.31 0.57
frieden 1.00 1.00
peter 0.50 0.50 0.50 0.48 0.46 1.00
gemuese 0.24 0.68 0.24 0.24 0.57 0.47 0.93 0.43
vegetarier 0.30 0.24 0.30 0.30 0.53 0.57 0.29 1.00
krieg 1.00 1.00

Back
Close

Evaluation: Cross-lingual thesauri



Conclusion

- Theoretical foundations
- Performance
- Improved precision
- Ease of integration
- Multiple use
- Cross-language retrieval

Handout, Sample code

The handout for this talk is available at

`http://sites.inka.de/moebius/docs/simthes-ho.pdf`

Sample code is available at

`http://sites.inka.de/moebius/comp/simthes/`

Thanks very much for your attendance!