Ontology-based Information-Filtering and Retrieval

BIS 2005

Dominik Kuropka

Some information about me…

till February 2004

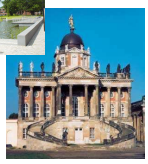
- research assistant
  - @ Char of Prof. Dr. Jörg Becker
  - @ Institute for Information Systems
  - @ University of Münster, Germany
- I wrote my doctoral thesis

  Models for representation of natural language documents – ontology based IF&IR with relational databases.

  (translated title)

Some information about me…

since July 2004



- research assistant
  - @ Char of Prof. Dr. Mathias Weske (Business Process Technology)
  - @ Hasso-Platter-Institute
  - @ University of Potsdam
- active member at the (EU funded) scientific project

  Adaptive Services Grid

Acknowledgement

- Prof. Dr. Jörg Becker and Prof. Dr. Ulrich Müller-Funk

- Prof. Dr. Mathias Weske

- Prof. Dr. Witold Abramowicz

Some information about you…

- Why are you here?
- What's your background?

Structure

Current Situation

- Around 1980 computers get small enough to be used on the desk
  - increase of office applications
  - increase of natural language documents processed and stored by computers
- Since 1990: World Wide Web
  - world-wide distributed documents can be accessed easily
  - publication of documents is very easy

=> Huge number of documents is accessible

Infoglut / Information Overload

Situation
- For problem solving and decision making information is essential
- Computers and networks make a huge amount of information available

Problem

Human capacity for information processing is limited

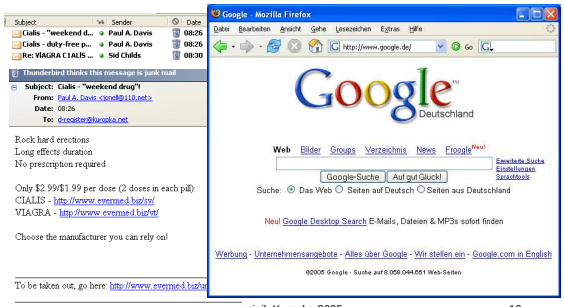
Result

not all information can be processed, relevant information may founder in the flood of information

Solution attempt

- Computer help on **filtering** and **retrieval** of relevant information
- Challenges
  - Which information is relevant?
  - Information is written/coded in natural language
  - Natural languages are ambiguous (in comparison to formal languages)
  - Natural languages have a high expressiveness
  - Many different natural languages exist

IF&IR in your daily life…



Ambiguity of natural languages

"time flies like an arrow"

Sense 1 (usually indented):

time (temporal) progresses fast

Sense 2:

"time flies" (some special flies) like/love an arrow

Sense 3:

quantify the speed of flies, like an arrow does it

Ontology

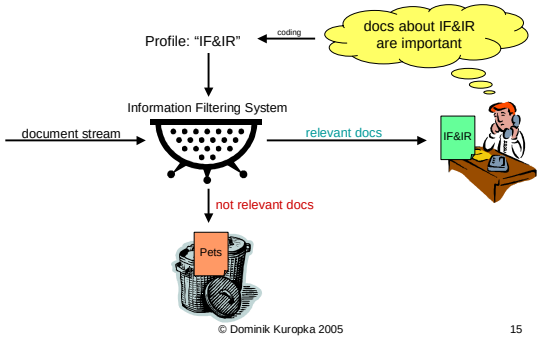
- An ontology is…
  - … an explicit specification of a conceptualization. [Gruber 1993, 1995]
  - … a model of linguistic means of expression on which several actors have agreed on and which are used by those actors.
- An ontology defines concepts and their relations, e.g.
  - an "arrow" is something pointing into a direction
  - "fly" = aviation or animal
  - "time fly" is <u>not</u> a valid special class of a fly
- Usage of an ontology has the potential
  - … to reduce ambiguity of natural languages
  - … to rise quality (prec. and recall) of IF&IR applications

Scope of this tutorial

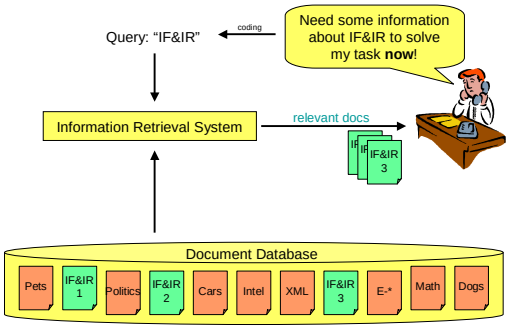
- Overview on information filtering and retrieval
  - basic terminology and demarcation
  - classification of popular approaches
- Presentation of two “bleeding edge” approaches
  - TVSM: is the fundament
  - eTVSM: ontology based approach for IF&IR

Structure

Information Filtering
from the user’s perspective



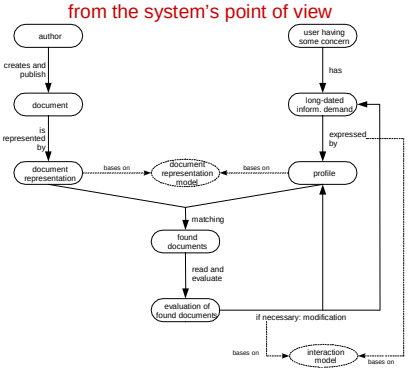
Information Retrieval
from the user’s perspective



Findings: IF vs. IR
from the user’s point of view

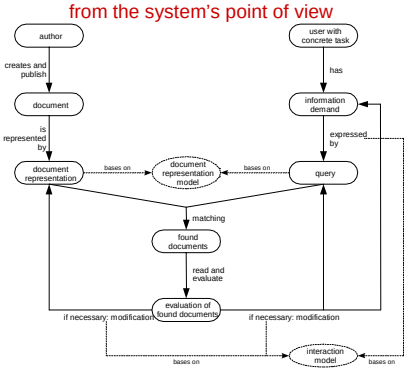
- Information Filtering (IF) is…
  the selection of documents
  from a dynamic stream of documents
  using some kind of static profile.
- Information Retrieval (IR) is…
  the selection of documents
  from a static set of documents
  using an ad-hoc query.
- Conclusion
  IF and IR are different tasks!

Information Filtering
from the system’s point of view



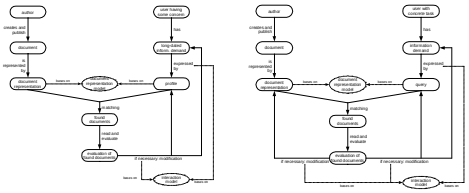
Information Retrieval

from the system's point of view



Findings: IF vs. IR

from the system's point of view



- IF and IR are structurally similar even though they are different tasks.

Document Representation Model

- Computer need formal, computable models
- Document Representation Models define…
  - …how documents are represented
  - …how user profiles / queries have to be specified
  - …the similarity between
    - two documents or
    - a document and a query

Approaches toward a Document Representation Model

Statistical Approach

- no syntax, semantics
- lack of knowledge
- pure number crunching
- highly efficient
- not very effective
- State-of-the-Art

TVSM / eTVSM

- no syntax, but some semantics
- some knowledge <= ontology
- efficient
- (in theory) effective
- seems feasible

main focus

Artificial Intelligence

- syntax, semantics
- huge knowledge base
- understand meaning
- (currently) not efficient
- (in theory) very effective
- not really feasible (at the moment)

Structure

Common properties of document representation models

- Documents are initially represented as a vector of terms usually without any formatting.

"The best known mouse species is the common house mouse." → vector = {the, best, known, mouse, …}

- This vector is transformed into an other, a model specific representation.
- Terms are often equivalent to words (sometimes word-lists).

Classification of document representation models

- Different classifications are existing
  - Most famous: classification regarding the mathematical foundation
- Here
  - Classification regarding the characteristics of the term representation

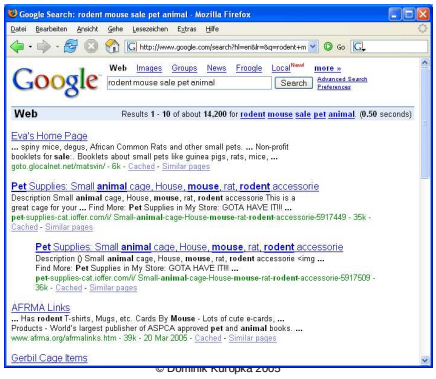
Models without term interdependencies

- Terms are assumed to be pair-wise independent / orthogonal
- most IF/IR systems fall into this class
  - Google, WebCrawler, all-the-web
  - Spam-Filter etc.

What are the implications of the independency assumption?

- Zero similarity between:
  - "*cars are fast*" and "*an automobile is quick*"
  - "*windows is cheap*" and "*microsoft closes a bug*"
- Non-Zero similarity between:
  - "*mouse and keyboard are necessary*" and "*my mouse likes cheese*"

Real-World Sample: Buy Mouse



Disadvantages of Models without term interdep.

- Linguistic relationships between terms, like
  - Inflection (e.g. house, houses)
  - Synonymy (e.g. car, automobile)
  - Homography (e.g. mouse [comp.], mouse [animal])
  - Hyponymy (e.g. plant, tulip)
  - Meronymy (e.g. wheel, car)
  - Composition (e.g. work, workaround)
  - Relevance of terms

  can <u>not</u> be represented

Workaround

- Document Preprocessing
  - Reduce variability of terms in documents
- Often used approaches
  - Stopword List (Relevance of terms)
  - Stemming (Inflection)
  - Synonymy Substitution (Synonymy)

Preprocessing example

This car is one of the *fastest* automobiles on earth.

remove formatting

this car is one of the fastest automobiles on earth

remove Stopwords

car one fastest automobiles earth

Stemming

car one fast automobile earth

Synonymy Subst.

car one fast car earth

Models with immanent term interdep.

- Terms may be pair-wise interdependent
- Term dependency is coupled with the model

- Grade of interdependency is derived using a co-occurrence based method
- Example for a simple co-occurrence based method:
  - "computer" occurs 345 times in document base
  - "linux" occurs 298 times in document base
  - "computer" and "linux" occurs 276 times in document base

  => "computer" and "linux" are highly interdependent

Problems with co-occurrence based term interdependencies

- In theory: Co-occurrences of terms is related to similarity of terms. (statistical approach)
- In theory: Linguistic phenomenons are related to similarity of terms. (linguistic approach) e.g.
  - Synonymy => high similarity
  - Inflection => high similarity
  - Word groups (e.g. New York) => low similarity

- In practice: Simple co-occurrences based similarities does not match linguistic based similarities

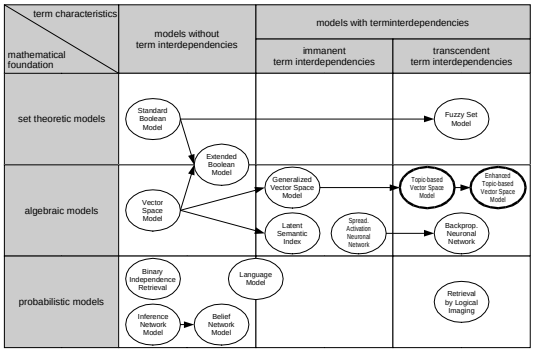
Co-occurrences of English Wikipedia (excerpt)

a	*b*	Jaccard	Dice	Cosine	Phenomenon	exp. Sim.
New	York	0,245	0,394	0,484	word group	very low
amber	nectar	0,005	0,010	0,011	word group	very low
car	tree	0,022	0,043	0,043	none	null
house	red	0,059	0,111	0,111	none	null
mastermind	master	0,004	0,008	0,025	composition	low
mastermind	mind	0,003	0,007	0,025	composition	low
wheel	car	0,061	0,115	0,135	meronymy	high
body	leg	0,023	0,045	0,080	meronymy	high
plant	tree	0,080	0,148	0,149	hyponymy	high
plant	tulip	0,005	0,010	0,035	hyponymy	high
car	automobile	0,142	0,249	0,278	synonymy	very high
hope	esperance	0,000	0,001	0,008	synonymy	very high
house	houses	0,072	0,135	0,179	inflection	very high
mouse	mice	0,108	0,195	0,226	inflection	very high

Models with transcendent term interdep.

- Terms may be pair-wise interdependent
- Term dependency is not coupled with the model
- Grade of dependency can be defined in an arbitrary manner
  - Opens IF/IR model for sophisticated methods of dependency derivation.
  - Manual definition of dependencies possible (e.g. by a given ontology)

Classification Overview



Evaluation of models regarding linguistic phenomenons

linguistic phenomenon \ IF/IR model		without term interdependencies						with immanent term interdep.			with transcendent term interdep.			
		SBM	VSM	EBM	BIR	INM	BNM	LM	GVSM	LSI	SANN	FSM	RbLI	BNN	TVSM	eTVSM
morphology	inflection	1	1	1	1	1	1	1	1	1	1	●	●	●	●	●
	composition	-	-	-	-	-	-	-	Ⓔ	Ⓔ	Ⓔ	●	●	●	●	●
	derivation	-	-	-	-	-	-	-	Ⓔ	Ⓔ	Ⓔ	●	●	●	●	●
lexical semantics	synonymy	2	2,4	2	2,4	2	2	2	Ⓔ	Ⓔ	Ⓔ	●	●	●	●	●
	homography	-	-	-	-	-	-	-	-	-	-	-	-	●	-	●
	metonymy	-	-	-	-	-	-	-	-	-	-	-	-	●	-	●
	hyponymy	-	4	-	4	-	-	-	Ⓔ	Ⓔ	Ⓔ	●	●	●	●	●
	meronymy	-	4	-	4	-	-	-	Ⓔ	Ⓔ	Ⓔ	●	●	●	●	●
	term weights	3	●	●	3	●	●	3 ○	3	3	●	3	●	●	●	●
word groups		-	○	-	-	-	-	○	-	-	-	-	-	-	-	●
comment													*A*	*B*		

Caption:

- ● = is represented
- Ⓔ = uses co-occurence based methods for representation
- ○ = add-ons to represent this phenomenon are existing
- - = is not represented
- 1 = only with stemming
- 2 = only with synonym substitution
- 3 = only with stopword list
- 4 = by query expansion approaches (usually co-occurrence based)
- *A* = model is not consistent
- *B* = needs huge amount of training data

Structure

Topic-based Vector Space Model

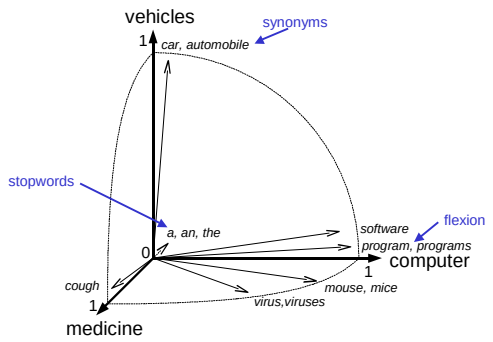
has following characteristics

- represents a lot of linguistic phenomenons (flexion, composition, derivation, synonymy, hyponymy, meronymy and term weights)
- explicit interface for term-interdependencies
- numerical complexity low enough for practical usage
- A proof for adequacy of usual preprocessing steps with the TVSM exists (this feature is not met by most other models)

Concept of the TVSM

- vector space based model
- queries/profiles are represented as virtual documents
- term weights are represented by vector length
- grades of term interdependencies are represented by angles between terms

Visualization of the TVSM concept



Linguistic phenomenons in relation to term angles

- Flexion (e.g. house and houses)
  maximal interdependency => term angles equals 0°
- Composition (e.g. master, mastermind)
  English: low interdependency => term angles tend to 90°
  German: high interdependency => term angles are low
- Derivation (e.g. industry, industrialization)
  high interdependency => term angles are low
- Synonymy (e.g. car, automobile)
  high interdependency => term angles are low or equals even 0°
- Hyponymy (e.g. plant, tree)
  high interdependency => term angles are low, depends on number of intermediate layers
- Meronymy (e.g. wheel, car)
  high interdependency => term angles are low, depends on number of intermediate layers

Mathematical representation of vector space as terms

- $d \in \mathbb{N}$ is the number topics = dimensions
- vector space: $R = \mathbb{R}^d_{\geq 0}$
- a term i is represented by a term-vector $\vec{t}_i$

$$\forall i \in T: \quad \vec{t}_i \in R \quad \wedge \quad |\vec{t}_i| = [0...1]$$

- angle between term i and j: $\omega_{i,j} \in [0°...90°]$

=> term-similarity: $\cos \omega_{i,j} \in [0...1]$

Mathematical representation of documents

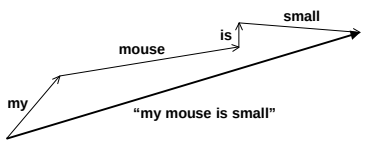
- set of all documents: D
- each document $k \in D$ is represented by a document-vector $\vec{d}_k \in R$

$$\forall k \in D: \quad \vec{d}_k = \frac{1}{|\vec{\delta}_k|}\vec{\delta}_k \quad \Rightarrow \quad |\vec{d}_k| = 1$$

with $\vec{\delta}_k = \sum_{i \in T} a_{k,i}\vec{t}_i$

and $a_{k,i}$ = occurrence of term i in document k

Visualization of the representation of documents



Mathematical representation of documents-similarities

- The similarity of two documents $k, l \in D$ is defined as

$$\begin{aligned} \mathrm{sim}(k,l) &= \vec{d}_k \vec{d}_l \\ &= |\vec{d}_k||\vec{d}_l| \cos \omega_{k,l} \\ &= \cos \omega_{k,l} \end{aligned}$$

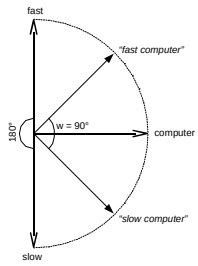
=> document-similarity = angle between document vectors

Properties of the document-similarity

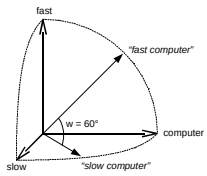
$\mathrm{sim}(k,l) \leq 1$	**because** $\cos \omega_{k,l} \leq 1$
$\mathrm{sim}(k,l) \geq 0$	**b.** $R = \mathbb{R}^d_{\geq 0} \Rightarrow \omega_{k,l} \in [0°; 90°]$
$\mathrm{sim}(k,l) = 1$	$\forall k - l$
$\mathrm{sim}(k,l) = \mathrm{sim}(l,k)$	$\forall k, l \in D$

Rationale for positive axis intercepts of R

a) negative and positive axis intercepts



b) only positive axis intercepts



Binding of attributes to an object may depend on the point of view:

A fast computer in 1995 is a slow computer in 2005!

Similarity: cos(180°) = –1
cos(90°) = 0
cos(60°) = 0.5

Computation of document similarities

Assumption: the following values are given

- set of all terms T
- set of all documents D
- occurrence of terms within documents $a_{k,i}$
- term weights given by term vector length $|\vec{t}_i|$
- angles between term-vectors $\omega_{i,j}$

Hint: term-vectors themselves are <u>not</u> needed
=> dim. d of vector-space is <u>not</u> relevant

Computation of document similarities

$$\vec{t}_i\vec{t}_j = |\vec{t}_i||\vec{t}_j| \cos \omega_{i,j}$$ ← weighted similarity of terms (scalar product)

$$|\vec{\delta}_k| = |\sum_{i \in T} a_{k,i}\vec{t}_i|$$ ← document vector length

$$= \sqrt{|\sum_{i \in T} a_{k,i}\vec{t}_i|^2}$$

$$= \sqrt{(\sum_{i \in T} a_{k,i}\vec{t}_i)^2}$$

$$= \sqrt{\sum_{i \in T}\sum_{j \in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j}$$

Computation of document similarities

$$\mathrm{sim}(k,l) = \vec{d}_k\vec{d}_l$$ ← similarity between documents

$$= \frac{1}{|\vec{\delta}_k|}\vec{\delta}_k\frac{1}{|\vec{\delta}_l|}\vec{\delta}_l$$

$$= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\vec{\delta}_k\vec{\delta}_l$$

$$= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i \in T} a_{k,i}\vec{t}_i \sum_{j \in T} a_{l,j}\vec{t}_j$$

$$= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i \in T}\sum_{j \in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j$$

Findings provided by the TVSM

- In contrast to other IF&IR models applicability of preprocessing steps like
  - stopword list
  - stemming
  - synonym substitution
- can be formally proved with the TVSM
  - ⇒Hidden assumptions of preprocessing are explicated

Stopword-Lemma

- Set of all terms: T
- Set of all stopwords: $T_\oslash$

- Assumption:
  stopwords in a stopword list are irrelevant for the relation of a document towards a topic

$$|\vec{t}_i| = 0 \qquad \forall i \in T_\oslash \subset T$$

$$\vec{t}_i\vec{t}_j = |\vec{t}_i||\vec{t}_j| \cos \omega_{i,j} = 0 \qquad \text{for} \quad i \in T_\oslash \ \vee \ j \in T_\oslash$$

Stopword-Lemma

$$|\vec{\delta}_k| = \sqrt{\sum_{i \in T}\sum_{j \in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j}$$

$$= \sqrt{\sum_{i \in T \setminus T_\oslash}\sum_{j \in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{i \in T_\oslash}\sum_{j \in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j}_{=0}}$$

$$= \sqrt{\sum_{i \in T \setminus T_\oslash}\left(\sum_{j \in T \setminus T_\oslash} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{j \in T_\oslash} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j}_{=0}\right)}$$

$$= \sqrt{\sum_{i \in T \setminus T_\oslash}\sum_{j \in T \setminus T_\oslash} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j}$$

Stopword-Lemma

$$\begin{aligned}
\mathrm{sim}(k,l) &= \vec{d}_k\vec{d}_l \\
&= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i\in T}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j \\
&= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i\in T\setminus T_\diamond}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{i\in T_\diamond}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j}_{=0} \\
&= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i\in T\setminus T_\diamond}\left(\sum_{j\in T\setminus T_\diamond} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j + \underbrace{\sum_{j\in T_\diamond} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j}_{=0}\right) \\
&= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i\in T\setminus T_\diamond}\sum_{j\in T\setminus T_\diamond} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j
\end{aligned}$$

Stopword-Lemma

Findings

- Stopword lists are compatible with the TVSM
- Assumption:
  - document topic is independent from stopwords
  - => stopwords have a weight value of null
- Assumption generally compatible to linguistics if stopwords are articles
  (like: the, is, a, an, …)

Stemming-Lemma

- Set of all terms: T
- Set of all stems: $T_\perp$
- Stemming function:

$$\perp : T \rightarrow T_\perp \qquad \perp^{-1} : T_\perp \rightarrow \wp(T)$$

$$\begin{array}{rl}
T_\perp \subseteq T & \\
\perp(i) \in T_\perp & \forall i \in T \\
\perp^{-1}(o) \subseteq T \ \wedge \ o \in \perp^{-1}(o) & \forall o \in T_\perp \\
\perp(i) = o & \forall o \in T_\perp,\ i \in \perp^{-1}(o) \\
\nexists i : \ i \in \perp^{-1}(o) \ \wedge \ i \in \perp^{-1}(p) & \forall o \neq p
\end{array}$$

Stemming-Lemma

- Assumption:
  - Relationship of a term to a topic is the same as the relationship of the stem of the term to the topic.
  - The weight of a term is equal to the weight of the stem of the term
- Formally:

$$\begin{array}{ll}
\omega_{i,o} = \omega_{o,i} = 0^\circ \quad \wedge \quad |\vec{t}_i| = |\vec{t}_o| & \forall i \in T,\ o = \perp(i) \\
\Rightarrow \quad \vec{t}_i = \vec{t}_o &
\end{array}$$

Stemming-Lemma

$$\begin{aligned}
|\vec{\delta}_k| &= \sqrt{\sum_{i\in T}\sum_{j\in T} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j} \\
&= \sqrt{\sum_{o\in T_\perp}\sum_{i\in\perp^{-1}(o)}\sum_{p\in T_\perp}\sum_{j\in\perp^{-1}(p)} a_{k,i}a_{k,j}\vec{t}_i\vec{t}_j} \\
&= \sqrt{\sum_{o\in T_\perp}\sum_{p\in T_\perp}\sum_{i\in\perp^{-1}(o)}\sum_{j\in\perp^{-1}(p)} a_{k,i}a_{k,j}\vec{t}_o\vec{t}_p} \\
&= \sqrt{\sum_{o\in T_\perp}\sum_{p\in T_\perp}\vec{t}_o\vec{t}_p\left(\sum_{i\in\perp^{-1}(o)}\sum_{j\in\perp^{-1}(p)} a_{k,i}a_{k,j}\right)} \\
&= \sqrt{\sum_{o\in T_\perp}\sum_{p\in T_\perp}\vec{t}_o\vec{t}_p\left(\sum_{i\in\perp^{-1}(o)} a_{k,i}\right)\left(\sum_{j\in\perp^{-1}(p)} a_{k,j}\right)} \\
&= \sqrt{\sum_{o\in T_\perp}\sum_{p\in T_\perp} a'_{k,o}a'_{k,p}\vec{t}_o\vec{t}_p}
\end{aligned}$$

with $a'_{k,o} = \sum_{i\in\perp^{-1}(o)} a_{k,i}$ and $a'_{k,p} = \sum_{j\in\perp^{-1}(p)} a_{k,j}$

Stemming-Lemma

$$\begin{aligned}
\vec{d}_k\vec{d}_l &= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i\in T}\sum_{j\in T} a_{k,i}a_{l,j}\vec{t}_i\vec{t}_j \\
&\ \ \vdots \\
&= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{o\in T_\perp}\sum_{p\in T_\perp} a'_{k,o}a'_{l,p}\vec{t}_o\vec{t}_p
\end{aligned}$$

with $a'_{k,o} = \sum_{i\in\perp^{-1}(o)} a_{k,i}$ and $a'_{l,p} = \sum_{j\in\perp^{-1}(p)} a_{l,j}$

Stemming-Lemma

Findings

- Stemming is compatible with the TVSM
- Assumption:
  - Relationship of a term to a topic is the same as the relationship of the stem of the term to the topic.
  - The weight of a term is equal to the weight of the stem of the term
- Depending on the language the assumption may be compatible to linguistics :
  - words and their stems are usually assigned to the same topic,
  - but the assignment of words to a stem is not unique for all languages!
    (e.g. German: “sucht” => Stems: “suchen” or “Sucht”)

Synonymy-Lemma

- Set of all terms: T
- Set of all leading synonyms: T_f
- Synonymy function:

$$F : T \rightarrow T_f \qquad F^{-1} : T_f \rightarrow \wp(T)$$

$$T_f \subseteq T$$

$$F(i) \in T_f \qquad \forall i \in T$$

$$F^{-1}(o) \subseteq T \ \wedge \ o \in F^{-1}(o) \qquad \forall o \in T_f$$

$$F(i) = o \qquad \forall o \in T_f,\ i \in F^{-1}(o)$$

$$\nexists i :\ i \in F^{-1}(o) \ \wedge \ i \in F^{-1}(p) \qquad \forall o \neq p$$

Synonymy-Lemma

- Assumption:
  - Relationship of a term to a topic is the same as the relationship of the leading term to the topic.
  - The weight of a term is equal to the weight of the leading term
  - Total Synonymy
- Formally:

$$\omega_{i,o} = \omega_{o,i} = 0^\circ \quad \wedge \quad |\vec{t}_i| = |\vec{t}_o| \qquad \forall i \in T,\ o = F(i)$$

$$\Rightarrow \qquad \vec{t}_i = \vec{t}_o$$

Synonymy-Lemma

$$|\vec{\delta}_k| = \sqrt{\sum_{i \in T} \sum_{j \in T} a_{k,i} a_{k,j} \vec{t}_i \vec{t}_j}$$

$$\vdots$$

$$= \sqrt{\sum_{o \in T_f} \sum_{p \in T_f} a'_{k,o} a'_{k,p} \vec{t}_o \vec{t}_p}$$

with $a'_{k,o} = \sum_{i \in F^{-1}(o)} a_{k,i}$ and $a'_{k,p} = \sum_{j \in F^{-1}(p)} a_{k,j}$

$$\vec{d}_k \vec{d}_l = \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|} \sum_{i \in T} \sum_{j \in T} a_{k,i} a_{l,j} \vec{t}_i \vec{t}_j$$

$$\vdots$$

$$= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|} \sum_{o \in T_f} \sum_{p \in T_f} a'_{k,o} a'_{l,p} \vec{t}_o \vec{t}_p$$

with $a'_{k,o} = \sum_{i \in F^{-1}(o)} a_{k,i}$ and $a'_{l,p} = \sum_{j \in F^{-1}(p)} a_{l,j}$

Synonymy-Lemma

Findings

- Synonymy substitution is compatible with the TVSM
- Assumption:
  - Relationship of a term to a topic is the same as the relationship of the leading term to the topic.
  - The weight of a term is equal to the weight of the leading term
  - Total Synonymy
- Assumption generally <u>not</u> compatible to linguistics:
  - When taking a broader context into account only Partial Synonymy can be observed
  - Example:
    - “rock” and “stone” are synonymous in most contexts
    - In the context of fruits “stone” is synonymous to a “hard seed”, but the term “rock” is not used/defined.

Pros and Cons of the TVSM

- Pros
  - It supports Flexion, Composition, Derivation, Synonymy, Hyponymy and Meronymy
  - It is compatible to Stopword Lists and Stemming
- Cons
  - Hints how term-similarities should be gained are not given
  - Homography and Metonymy are not supported
  - Word groups are not supported

Structure

1 Motivation
2 Information Filtering vs. Information Retrieval
3 Classification of popular IF&IR models
4 Topic-based Vector Space Model (TVSM)
5 Enhanced TVSM (eTVSM)
6 Application of the eTVSM approach

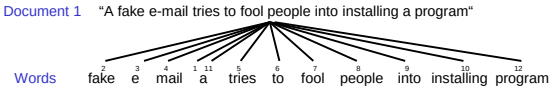
Enhanced TVSM

has following characteristics

- It bases on the TVSM
- Additionally, it is able to represent Homography, Metonymy and Word Groups
- It specifies formally how term/topic-similarities are derived from given topic-maps
- Integrates Stopword List and Stemming to rise calculation performance

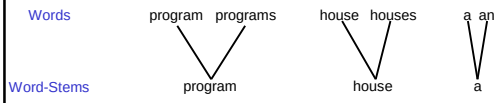
Concept of the eTVSM

Words are assigned to Documents, the assignment includes their position



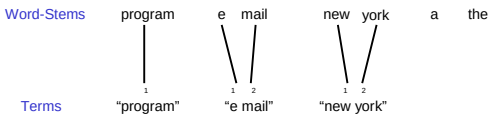
Concept of the eTVSM

- Words are assigned to Word-Stems
- This assigned can be derived from common stemming algorithms



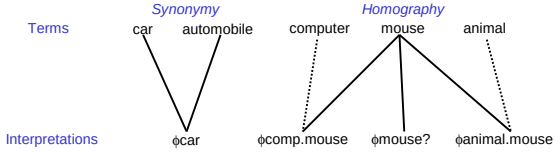
Concept of the eTVSM

- Word-Stems are assigned to Terms
  - Word Groups can be also a term
  - Not all Word-Stems have to be assigned (especially Stopwords should be not assigned)



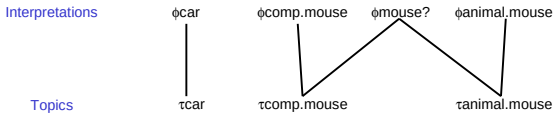
Concept of the eTVSM

- Terms are assigned to Interpretations as a valid representation of an Interpretation
- Further, Terms may be assigned to Interpretations as so called Support-Terms

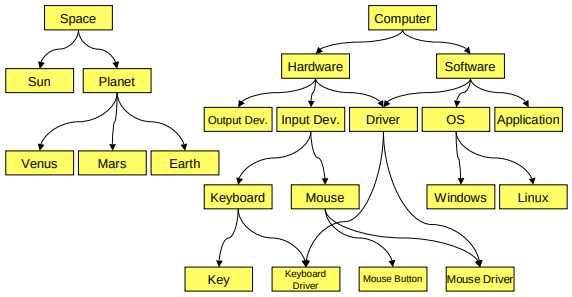


Concept of the eTVSM

- Interpretations are assigned to Topics
- Topics are represented by vectors
- Topics are the fundament of the eTVSM
  (like terms where the fundament of the TVSM)



A sample Topic Map



Sub-Goal: get Topic Similarities (1/2)

	Space	Sun	Planet	Venus	Earth	Mars	Computer	Hardware	Output Dev.	Input Dev.	Keyboard
Space	1,000	0,855	0,855	0,754	0,754	0,754	0,000	0,000	0,000	0,000	0,000
Sun	0,855	1,000	0,463	0,408	0,408	0,408	0,000	0,000	0,000	0,000	0,000
Planet	0,855	0,463	1,000	0,882	0,882	0,882	0,000	0,000	0,000	0,000	0,000
Venus	0,754	0,408	0,882	1,000	0,667	0,667	0,000	0,000	0,000	0,000	0,000
Earth	0,754	0,408	0,882	0,667	1,000	0,667	0,000	0,000	0,000	0,000	0,000
Mars	0,754	0,408	0,882	0,667	0,667	1,000	0,000	0,000	0,000	0,000	0,000
Computer	0,000	0,000	0,000	0,000	0,000	0,000	1,000	0,931	0,659	0,885	0,815
Hardware	0,000	0,000	0,000	0,000	0,000	0,000	0,931	1,000	0,774	0,947	0,872
Output Dev.	0,000	0,000	0,000	0,000	0,000	0,000	0,659	0,774	1,000	0,565	0,520
Input Dev.	0,000	0,000	0,000	0,000	0,000	0,000	0,885	0,947	0,565	1,000	0,920
Keyboard	0,000	0,000	0,000	0,000	0,000	0,000	0,815	0,872	0,520	0,920	1,000
Key	0,000	0,000	0,000	0,000	0,000	0,000	0,652	0,753	0,516	0,826	0,915
Keyboard Driv.	0,000	0,000	0,000	0,000	0,000	0,000	0,839	0,844	0,436	0,860	0,915
Mouse	0,000	0,000	0,000	0,000	0,000	0,000	0,815	0,872	0,520	0,920	0,695
Mouse Button	0,000	0,000	0,000	0,000	0,000	0,000	0,652	0,753	0,516	0,826	0,605
Mouse Driv.	0,000	0,000	0,000	0,000	0,000	0,000	0,839	0,844	0,436	0,860	0,667
Driver	0,000	0,000	0,000	0,000	0,000	0,000	0,906	0,912	0,471	0,928	0,855
Software	0,000	0,000	0,000	0,000	0,000	0,000	0,931	0,733	0,453	0,700	0,645
OS	0,000	0,000	0,000	0,000	0,000	0,000	0,678	0,428	0,309	0,382	0,351
Windows	0,000	0,000	0,000	0,000	0,000	0,000	0,635	0,400	0,289	0,357	0,329
Linux	0,000	0,000	0,000	0,000	0,000	0,000	0,635	0,400	0,289	0,357	0,329
Application	0,000	0,000	0,000	0,000	0,000	0,000	0,704	0,462	0,333	0,412	0,379

Sub-Goal: get Topic Similarities (2/2)

	Key	Keyboard Driv.	Mouse	Mouse Button	Mouse Driver	Driver	Software	OS	Windows	Linux	Application
Space	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Sun	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Planet	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Venus	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Earth	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Mars	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Computer	0,652	0,839	0,815	0,652	0,839	0,906	0,931	0,678	0,635	0,635	0,704
Hardware	0,753	0,844	0,872	0,753	0,844	0,912	0,733	0,428	0,400	0,400	0,462
Output Dev.	0,516	0,436	0,520	0,516	0,436	0,471	0,453	0,309	0,289	0,289	0,333
Input Dev.	0,826	0,860	0,920	0,826	0,860	0,928	0,700	0,382	0,357	0,357	0,412
Keyboard	0,915	0,915	0,695	0,605	0,667	0,855	0,645	0,351	0,329	0,329	0,379
Key	1,000	0,676	0,605	0,600	0,507	0,639	0,462	0,239	0,224	0,224	0,258
Keyboard Driv.	0,676	1,000	0,667	0,507	0,714	0,926	0,718	0,404	0,378	0,378	0,436
Mouse	0,605	0,667	1,000	0,915	0,915	0,855	0,645	0,351	0,329	0,329	0,379
Mouse Button	0,600	0,507	0,915	1,000	0,676	0,639	0,462	0,239	0,224	0,224	0,258
Mouse Driv.	0,507	0,714	0,915	0,676	1,000	0,926	0,718	0,404	0,378	0,378	0,436
Driver	0,639	0,926	0,855	0,639	0,926	1,000	0,776	0,436	0,408	0,408	0,471
Software	0,462	0,718	0,645	0,462	0,718	0,776	1,000	0,835	0,781	0,781	0,849
OS	0,239	0,404	0,351	0,239	0,404	0,436	0,835	1,000	0,935	0,935	0,617
Windows	0,224	0,378	0,329	0,224	0,378	0,408	0,781	0,935	1,000	0,750	0,577
Linux	0,224	0,378	0,329	0,224	0,378	0,408	0,781	0,935	0,750	1,000	0,577
Application	0,258	0,436	0,379	0,258	0,436	0,471	0,849	0,617	0,577	0,577	1,000

Approach to gain Topic Similarities

Two Steps

- Gain Topic Vectors from the Topic Map
- Calculate Topic Similarities from the scalar product of Topic Vectors

This approach bases on a formal representation of Topic Maps.

Formal representation of Topic Maps

Topic Maps will be represented here by

- a set of all topics: $\Theta = \{\tau_1, \tau_2, ..., \tau_{\#\Theta}\}$
- a super topic relation: $S(\tau_i) \subseteq (\Theta \setminus \tau_i)$

Derivations

- the transitive super topic relation for level p:

$$S^p(\tau_i) = S(\tau_i) \quad \text{for} \quad p = 1$$

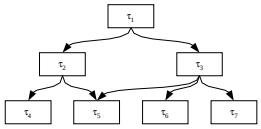
$$S^p(\tau_i) = \bigcup_{\tau_k \in S^{p-1}(\tau_i)} S(\tau_k) \quad \text{for} \quad p > 1$$

- the transitive unbounded super topic relation:

$$S^*(\tau_i) = S^1(\tau_i) \cup S^2(\tau_i) \cup S^3(\tau_i) \cup ...$$

Abstract Sample

$\Theta=\{\tau_1,\tau_2,\tau_3,\tau_4,\tau_5,\tau_6,\tau_7\}$
$S^*(\tau_1)=\{\}$
$S^*(\tau_2)=\{\tau_1\}$
$S^*(\tau_3)=\{\tau_1\}$
$S^*(\tau_4)=\{\tau_1,\tau_2\}$
$S^*(\tau_5)=\{\tau_1,\tau_2,\tau_3\}$
$S^*(\tau_6)=\{\tau_1,\tau_3\}$
$S^*(\tau_7)=\{\tau_1,\tau_3\}$



How to gain Topic Vectors from a formal Topic Map?

For topic leaves

- definition of topic leaves:

$$\Theta_B = \{\tau_i \in \Theta : \nexists \tau_k \in \Theta \text{ with } \tau_i \in S(\tau_k)\}$$

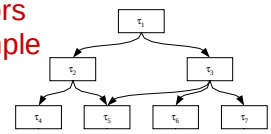
- abstract sample: $\Theta_B = \{\tau_4, \tau_5, \tau_6, \tau_7\}$

- definition of Topic Vectors:

$$\forall \tau_i \in \Theta_B : \vec{\tau}_i = |(\tau^*_{i,1}, \tau^*_{i,2}, ..., \tau^*_{i,\#\Theta})|$$

with $\tau^*_{i,d} = \begin{cases} 1 & \text{if} \quad \tau_d \in S^*(\tau_i) \ \vee \ i = d \\ 0 & \text{else} \end{cases}$

Topic Leave Vectors for the abstract sample



$$\tau_4 = |(1;1;0;1;0;0;0)| = (\frac{1}{\sqrt{3}};\frac{1}{\sqrt{3}};0;\frac{1}{\sqrt{3}};0;0;0)$$

$$\tau_5 = |(1;1;1;0;1;0;0)| = (\frac{1}{2};\frac{1}{2};\frac{1}{2};0;\frac{1}{2};0;0)$$

$$\tau_6 = |(1;0;1;0;0;1;0)| = (\frac{1}{\sqrt{3}};0;\frac{1}{\sqrt{3}};0;0;\frac{1}{\sqrt{3}};0)$$

$$\tau_7 = |(1;0;1;0;0;0;1)| = (\frac{1}{\sqrt{3}};0;\frac{1}{\sqrt{3}};0;0;0;\frac{1}{\sqrt{3}})$$

How to gain Topic Vectors from a formal Topic Map?

For topic nodes

- definition of topic nodes

$$\Theta_K = \complement\Theta_B = \Theta \setminus \Theta_B$$

- definition Topic Vectors

$$\forall \tau_i \in \Theta_K : \vec{\tau}_i = |\sum_{\tau_s \in \Theta:\ \tau_i \in S(\tau_s)} \vec{\tau}_s|$$

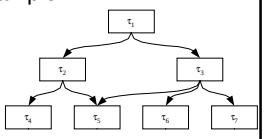
Topic Similarities

- Definition:

$$\begin{aligned} \mathrm{sim}(\tau_a, \tau_b) &= \vec{\tau}_a \vec{\tau}_b \\ &= \sum_{i=1}^{\#\Theta} \tau_{a,i}\tau_{b,i} \\ &= \cos\omega_{a,b} \end{aligned}$$

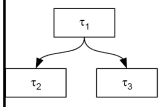
- Values for the abstract sample

	1	2	4	5	3	6	7
1	1,000	0,933	0,734	0,924	0,933	0,741	0,741
2	0,933	1,000	0,888	0,888	0,742	0,513	0,513
4	0,734	0,888	1,000	0,577	0,483	0,333	0,333
5	0,924	0,888	0,577	1,000	0,836	0,577	0,577
3	0,933	0,742	0,483	0,836	1,000	0,871	0,871
6	0,741	0,513	0,333	0,577	0,871	1,000	0,667
7	0,741	0,513	0,333	0,577	0,871	0,667	1,000

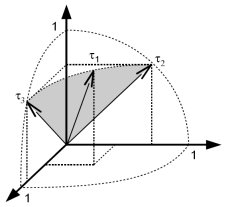


Illustration

Topic Map



Topic Vectors



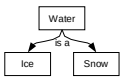
Topic Similarities

	τ_1	τ_2	τ_3
τ_1	1,000	0,866	0,866
τ_2	0,866	1,000	0,500
τ_3	0,866	0,500	1,000

Topic Map Modeling Issues

Topic Maps | Topic Similarities

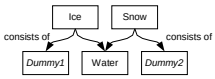
a)



	Water	Ice	Snow
Water	1,000	0,866	0,866
Ice	0,866	1,000	0,500
Snow	0,866	0,500	1,000

sim(Ice+Snow, Water) = 1,000

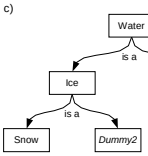
b)



	Water	Ice	Snow
Water	1,000	0,839	0,839
Ice	0,839	1,000	0,644
Snow	0,839	0,644	1,000

sim(Ice+Snow, Water) = 0,925

c)



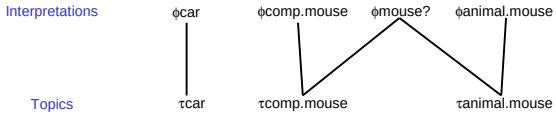
	Water	Ice	Snow
Water	1,000	0,851	0,777
Ice	0,851	1,000	0,913
Snow	0,777	0,913	1,000

sim(Ice+Snow, Water) = 0,832

Interpretation Similarities

Remember:

Interpretations are assigned to topics



Question:

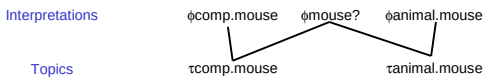
How are Interpretation Vectors and Similarities gained?

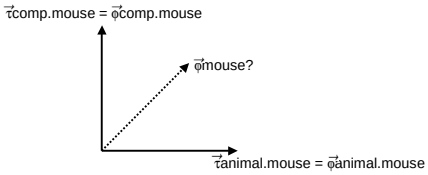
Gaining Interpretation Vectors

- Formalism:
  - Interpretation, set of all Interpretations: $\phi_i \in \Phi$
  - Interpretation Weight: $g(\phi_i) \in [0...1]$
  - Interpretation to Topic assignment: $T(\phi_i) \in \wp(\Theta) \setminus \{\}$
  - Topic Vectors: $\vec{\tau}_k$
- Definition of Interpretation Vectors:

$$\vec{\phi}_i = \frac{g(\phi_i)}{|\sum_{\tau_k \in T(\phi_i)} \vec{\tau}_k|} \cdot \sum_{\tau_k \in T(\phi_i)} \vec{\tau}_k$$

Interpretation Vectors





Interpretation Similarities

Definition: the similarity of two interpretations is equal to the scalar product of the interpretations:

$$\vec{\phi}_i \vec{\phi}_k = \sum_{n=1}^{\#\Theta} \phi_{i,n} \phi_{k,n}$$

with $\vec{\phi}_i = (\phi_{i,1}, \phi_{i,2}, ..., \phi_{i,\#\Theta})$

=> The angle between interpretation vectors is

$$\omega_{i,k} = \cos^{-1} \frac{\vec{\phi}_i \vec{\phi}_k}{g(\phi_i) g(\phi_k)}$$

Document Similarities

- Document Similarities are derived from Interpretation Similarities
- Definition: Document Vector

$$\forall k \in D: \quad \vec{d}_k = \frac{1}{|\vec{\delta}_k|} \vec{\delta}_k \quad \Rightarrow \quad |\vec{d}_k| = 1$$

with $\vec{\delta}_k = \sum_{i \in T} a_{k,i} \vec{\phi}_i$

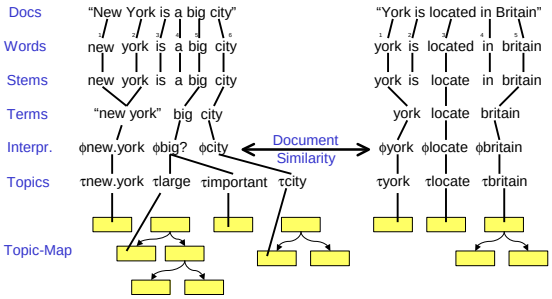
Calculation of Document Similarities

$$|\vec{\delta}_k| = |\sum_{i \in \Phi} a_{k,i}\vec{\phi}_i|$$
$$= \sqrt{|\sum_{i \in \Phi} a_{k,i}\vec{\phi}_i|^2}$$
$$= \sqrt{(\sum_{i \in \Phi} a_{k,i}\vec{\phi}_i)^2}$$
$$= \sqrt{\sum_{i \in \Phi}\sum_{j \in \Phi} a_{k,i}a_{k,j}\vec{\phi}_i\vec{\phi}_j}$$

$$\mathrm{sim}(k,l) = \vec{d}_k\vec{d}_l$$
$$= \frac{1}{|\vec{\delta}_k|}\vec{\delta}_k\frac{1}{|\vec{\delta}_l|}\vec{\delta}_l$$
$$= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\vec{\delta}_k\vec{\delta}_l$$
$$= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i \in \Phi} a_{k,i}\vec{\phi}_i \sum_{j \in \Phi} a_{l,j}\vec{\phi}_j$$
$$= \frac{1}{|\vec{\delta}_k||\vec{\delta}_l|}\sum_{i \in \Phi}\sum_{j \in \Phi} a_{k,i}a_{l,j}\vec{\phi}_i\vec{\phi}_j$$

Calculation is analogue to the TVSM

Summary of the eTVSM approach



Comparison to the Vector Space Model by example

1. Torvalds schreibt an SCO.
2. McBride warnt die Open-Source-Gemeinde.
3. Windows hat Preisvorteile gegenüber Linux.
4. Microsoft schließt Sicherheitslücken.
5. Neue Bugs in Windows.
6. Mit Maus und Tastatur geht es leichter.
7. Mäuse leben gerne in Löchern.

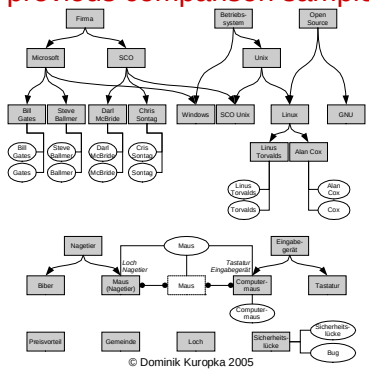
Dokumentenähnlichkeiten: eTVSM

	1	2	3	4	5	6	7
1	1,000	0,660	0,615	0,245	0,292	0,000	0,000
2	0,660	1,000	0,387	0,174	0,177	0,000	0,000
3	0,615	0,387	1,000	0,352	0,469	0,000	0,000
4	0,245	0,174	0,352	1,000	0,918	0,000	0,000
5	0,292	0,177	0,469	0,918	1,000	0,000	0,000
6	0,000	0,000	0,000	0,000	0,000	1,000	0,000
7	0,000	0,000	0,000	0,000	0,000	0,000	1,000

Dokumentenähnlichkeiten: VSM

	1	2	3	4	5	6	7
1	1,000	0,000	0,000	0,000	0,000	0,000	0,000
2	0,000	1,000	0,000	0,000	0,000	0,000	0,000
3	0,000	0,000	1,000	0,000	0,224	0,000	0,000
4	0,000	0,000	0,000	1,000	0,000	0,000	0,000
5	0,000	0,000	0,224	0,000	1,000	0,000	0,000
6	0,000	0,000	0,000	0,000	0,000	1,000	0,293
7	0,000	0,000	0,000	0,000	0,000	0,293	1,000

Ontology (Topic Map) for the previous comparison sample



Summary: Features of the eTVSM

- Is an advanced Vector Space Model
- Stopword removal and Stemming are integrated
- Can represent many linguistic phenomenons by term similarities
  - Word groups
  - Synonymy, Homonymy
  - Relations like Meronymy and Hyponymy
- Includes a heuristics to transform Topic-Maps into term similarities.

Criticism on the eTVSM

- Computational complexity of eTVSM depends on the composition of the topic-map
  - Worst case: $O(n*m)$ [all terms are related]
  - Best case: $O(\min(n,m))$ [all terms are orthogonal]
  (while n, m number of different terms in compared documents)
- eTVSM is currently "just" theory
  - Experiments with large document sets have to be provided in the future
  - But simple examples are promising
- Topic-Maps are needed for the application of eTVSM
  - This issue will be addressed on the next slides…

Structure

1 Motivation
2 Information Filtering vs. Information Retrieval
3 Classification of popular IF&IR models
4 Topic-based Vector Space Model (TVSM)
5 Enhanced TVSM (eTVSM)
6 Application of the eTVSM approach

How the get a Topic Map?

Possible strategies
- Creation of a new topic map (ontology) from scratch
  - Big bang approach
  - Gradual approach
  - Automated approach
- Reuse of an existing ontology

Big bang approach

Idea:

A group of users creates an ontology form the scratch before the IF/IR system is online.

Pros:
- It is easier to ensure consistency
- Errors are easier to avoid

Cons:
- System is not usable for a long time

Gradual approach

Idea:

eTVSM is "backward compatible" to the VSM if all interpretations are modeled as orthogonal. Start with a trivial ontology where interpretations are not related. Add relations betweens them on demand.

Pros:
- System is usable from the beginning
- It is possible to measure benefits of ontology modification

Cons:
- System quality is as low as normal VSM at the beginning
- Risk of inconsistencies within the ontology.

Automated approach

Idea:

Find statistical approaches to automate ontology creation. E.g. *Text-to-Onto* Tool
[A. Maedche: Ontology Learning for the Semantic Web. Kluwer Academic Publishers, 2002.]

Pros:
- Reduces manual work
- Ontology is optimized towards the document base

Cons:
- Currently a "bleeding edge" technology.

Reuse of an existing ontology

Idea:

Take an existing ontology (e.g. *WordNet*) and transform it to a adequate Topic Map.

Pros:
- Reduces manual work
- Existing ontology is usually field-tested

Cons:
- Conversion problems may occur

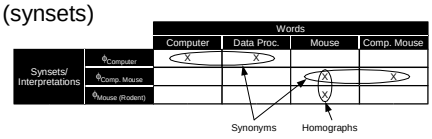
Existing Ontology: *WordNet*

- Initiated in 1985 by psychologists and linguists at the University of Princeton.
- Available for free at http://wordnet.princeton.edu/
- Current number of words: 203145
- Supported word classes:
  - nouns, verbs, adjectives, adverbs.
- Supported linguistic phenomenons:
  - synonymy, homography/polysemy, antonymy, hyponymy and meronymy.

Concept of the *WordNet*

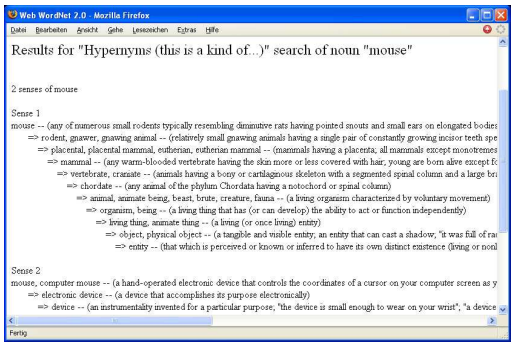
- Observation
  - One word may have several different senses
  - Several different words may have the same sense

=> Words are assigned to "synonymous sets" (synsets)



Synsets in WordNet ≈ Interpretations in eTVSM

WordNet: Sample



Potential of eTVSM for Information Retrieval

Types of Searches

- Navigational Searches
  - User knows exactly which document he wants, but he does not know how/where to get it.
  - E.g.: "Semantic Search Guha McCool Miller"
  - Well supported by current IR-Tools (e.g. google)
  - Plain word matching is adequate
- Research Searches
  - Users has a topic of interest, but he does not know a relevant document. Goal: Find a document matching users topic of interest.
  - E.g.: "Information Retrieval", "Cheap Cars"
  - Currently not very well supported by existing IR-Tools due to lack of semantics
  - eTVSM has potential to provide significant improvements

Application of eTVSM for Information Retrieval

1. Normal search strategy
   - User-Query and presentation of relevant Documents as usual
   - Implicit usage of Topic-Map/Ontology
2. Guided search strategy
   - Ask user for more precise specifications in case of not-resolvable homographs.
   - Propose adequate Super- or Subtopics to guide the user.

Challenges of Information Filtering

- Explication of long-dated information needs in the shape of a user profile is a hard job!
- Reasons:
  - Complexity and size
    - User profile has to specify current and future documents correctly.
  - Effort vs. utility
    - Total utility of an IF system is maximal, if effort in programming the user profile is low and the quota of correct classified documents is high.
  - Internalized rules
    - Humans are often not able to explain in general how they evaluate the relevance (class affiliation) of a document.
  - Natural language phenomenons
    - Synonymy, Homography, etc.

Adaptive Approaches for IF

- Learn a user profile by
  - given training data or
  - user feedback
- Problems with linguistic phenomenons
  - E.g. Synonymy, Hyponymy: Each synonym, and each hyponymy relation has to be learned independently
    => unnecessary large user profile reduces adaptability
  - E.g. Homography: Learning algorithm may get confused or needs long time to learn
    => unnecessary large user profile and long learning phase.
- Use of eTVSM in combination with an adequate Ontology has the potential to reduce these problems.

Future Work on the eTVSM

Current state:

the eTVSM has been designed and evaluated only from theoretical perspective.

=> Need for practical / statistical evaluation.

Example:

- eTVSM versus VSM
- eTVSM versus LSI
- eTVSM versus …

Evaluation Criteria for IF&IR systems

- Efficiency measures
  - Cost: e.g. acquisition, operating and maintenance
  - Performance: e.g. duration for query processing (IR) or delay until relevant documents are forwarded to the user (IF)
  - Usability: effort need from user point of view, adequacy of result presentation, …

Evaluation Criteria for IF&IR systems

- Effectivity measures (for one test-case)
  - Precision: $\frac{\#(docs_{really}^{relevant} \cap docs_{system}^{relevant})}{\# docs_{system}^{relevant}}$
  - Recall: $\frac{\#(docs_{really}^{relevant} \cap docs_{system}^{relevant})}{\# docs_{really}^{relevant}}$
  - Error ratio: $\frac{\#(docs_{really}^{relevant} \cap docs_{system}^{non-rel.}) + \#(docs_{really}^{non-rel.} \cap docs_{system}^{relevant})}{\# docs}$

Evaluation Criteria for IF&IR systems

- Aggregation over several test-cases
  - makro evaluation
    - arithmetic mean over precision or recall values
  - micro evaluation:
    - mean over all document numbers used within precision or recall calculation

Summary

- eTVSM has the potential to be more effective in IF&IR than other approaches, because it is able to resolve linguistic phenomenons by the use of an ontology.
- eTVSM can be used with a trivial ontology which will be enhance step by step.
- Reusing of existing ontologies like *WordNet* seems possible.
- Extensive statistical evaluations of eTVSM are missing and has to be done in the future
  => There is much potential for your research!